

A Small Electronic Structure Problem

Mariana Rossi and Alan Lewis

March 2021

Contents

1	Matrix Diagonalization	2
1.1	Casting the problem for a simple matrix	2
1.2	Generalization for $n \times n$ Matrices	3
1.2.1	Example	3
1.3	Jacobi Rotation Algorithm	4
2	Numerical and Analytical Integration	6
2.1	Integrals for Hartree-Fock Calculations	6
2.2	Basis sets	7
2.3	Numerical Integration of Hydrogenic Basis Sets	8
2.4	Analytical Integration of Gaussian Basis Sets	10
2.4.1	Sneaky Tricks	10
2.4.2	Evaluating the Overlap Integrals	12
3	Hartree-Fock Implementation	15
3.1	Definitions	15
3.2	Closed-Shell Restricted Hartree-Fock	15
3.3	Linear Combination of Atomic Orbitals	17
3.4	Self-Consistent Field Approach	19
A	Integrals of GTOs	20
A.1	Overlap Integrals	20
A.2	Kinetic Energy Integrals	21
A.3	Nuclear Attraction Integrals	21
A.4	Electron Repulsion Integrals	22

1 Matrix Diagonalization

The diagonalization of symmetric matrices is a central component of most electronic structure methods, including Hartree Fock. The diagonalization of a matrix can be achieved by many different algorithms, and is typically a very complicated problem. Many packages exist to perform this operation. However, for some (small) problems, it is possible to program a relatively simple matrix diagonalization algorithm, which is helpful to understand the complexity and the importance of this operation. Here we will introduce and then program one such technique, namely Jacobi rotations.

1.1 Casting the problem for a simple matrix

When diagonalizing, the equation that we want to solve is normally called an eigenvalue problem, and can be written as

$$\mathbf{h}\mathbf{C} = \epsilon\mathbf{C}, \quad (1)$$

where \mathbf{h} is the matrix we wish to diagonalize, \mathbf{C} is the matrix of eigenvectors, and ϵ is the diagonal matrix of eigenvalues, $\epsilon_{ij} = \epsilon_{ii}\delta_{ij}$. Multiplying Eq. (1) from the left by \mathbf{C}^{-1} , we obtain

$$\mathbf{C}^{-1}\mathbf{h}\mathbf{C} = \epsilon. \quad (2)$$

If \mathbf{h} is a Hermitian matrix (i.e. it has real eigenvalues), \mathbf{C} can be chosen to be a unitary matrix; that is:

$$(\mathbf{C}^*)^T = \mathbf{C}^{-1}. \quad (3)$$

For this course we will further restrict ourselves to real matrices, for simplicity.

To obtain a simple solution to Eq. (1) we start by considering the simplest possible case of 2×2 matrices. A real orthogonal 2×2 matrix only has one degree of freedom, as we have three constraints: the normalization and orthogonality of the two columns. The most convenient way to incorporate this information is by choosing the matrix \mathbf{C} to be a rotation matrix about an angle θ :

$$C = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}, \quad (4)$$

with $c = \cos \theta$ and $s = \sin \theta$. We now require a value of θ such that the equation

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} \epsilon_{11} & 0 \\ 0 & \epsilon_{22} \end{pmatrix} \quad (5)$$

has a solution. Solving for $\epsilon_{12} = 0$ gives

$$h_{12}(c^2 - s^2) + sc(h_{22} - h_{11}) = 0. \quad (6)$$

We can now use some trigonometric identities to define

$$(c^2 - s^2) = \cos 2\theta = C \quad (7)$$

$$2sc = \sin 2\theta = S, \quad (8)$$

which allows us to rewrite Eq. (6) as

$$h_{12}C + \frac{1}{2}S(h_{22} - h_{11}) = 0, \quad (9)$$

and obtain the solution for θ as

$$\theta = \frac{1}{2} \arctan \frac{S}{C} = \frac{1}{2} \arctan \frac{2h_{12}}{h_{11} - h_{22}}. \quad (10)$$

1.2 Generalization for $n \times n$ Matrices

For larger matrices, it is clear that finding a matrix \mathbf{C} which eliminates one off-diagonal element h_{ij} using the procedure outlined above is not a complete solution, as each elimination creates new off-diagonal elements at other places. However, if the procedure is continued – eliminating each off-diagonal element in turn – it is found that the matrix \mathbf{h} can be brought into a diagonal form. Traditionally, the largest off-diagonal element was eliminated at each step until the matrix was diagonalized to some numerical tolerance. However, in practice the time taken to locate the largest off-diagonal element of a large matrix is much more than that required to eliminate elements, so instead one simply sweeps through the all of the off-diagonal elements, eliminating them in order until they are all below some tolerance.

The matrix \mathbf{C} which solves the eigenvalue problem is given by the product of all the individual “ 2×2 ” matrices used in this procedure. Of course, all these individual matrices are in fact $n \times n$ matrices, but they only have the four non-zero elements required to eliminate a single element of \mathbf{h} as described above, and 1 on all other diagonal elements.

1.2.1 Example

To illustrate the algorithm described above, we will now show the first few steps of the procedure to diagonalize a 3×3 matrix. We will use the following notation: \mathbf{h} is the matrix to be diagonalized, ϵ_j is the approximation to the diagonal matrix ϵ after j iterations of the algorithm, \mathbf{U}_j is the transformation matrix used in the j -th iteration, and \mathbf{C}_j is the approximation to the eigenvector matrix \mathbf{C} after j iterations, given by $\mathbf{C}_j = \Pi_{i=1}^j \mathbf{U}_i$.

Initial Values

$$\mathbf{h} = \begin{pmatrix} 1.00 & 4.00 & 5.00 \\ 4.00 & 2.00 & 6.00 \\ 5.00 & 6.00 & 3.00 \end{pmatrix} = \epsilon_0, \quad \mathbf{C}_0 = \begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$$

Step 1 (eliminating h_{12} and h_{21})

$$\mathbf{U}_1 = \begin{pmatrix} 0.75 & 0.66 & 0.00 \\ -0.66 & 0.75 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$$

$$\mathbf{C}_1 = \mathbf{U}_1 \cdot \mathbf{C}_0 = \begin{pmatrix} 0.75 & 0.66 & 0.00 \\ -0.66 & 0.75 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$$

$$\epsilon_1 = \mathbf{U}_1^T \epsilon_0 \mathbf{U}_1 = \mathbf{C}_1^T \mathbf{h} \mathbf{C}_1 = \begin{pmatrix} -2.53 & 0.00 & -0.22 \\ 0.00 & 5.53 & 7.81 \\ -0.22 & 7.81 & 3.00 \end{pmatrix}$$

Step 2 (eliminating h_{13} and h_{31})

$$\mathbf{U}_2 = \begin{pmatrix} 1.00 & 0.00 & -0.04 \\ 0.00 & 1.00 & 0.00 \\ 0.04 & 0.00 & 1.00 \end{pmatrix}$$

$$\mathbf{C}_2 = \mathbf{U}_2 \cdot \mathbf{C}_1 = \begin{pmatrix} 0.75 & 0.66 & -0.03 \\ -0.66 & 0.75 & 0.03 \\ 0.04 & 0.00 & 1.00 \end{pmatrix}$$

$$\boldsymbol{\epsilon}_2 = \mathbf{U}_2^T \boldsymbol{\epsilon}_1 \mathbf{U}_2 = \mathbf{C}_2^T \mathbf{h} \mathbf{C}_2 = \begin{pmatrix} -2.54 & 0.31 & 0.00 \\ 0.31 & 5.53 & 7.80 \\ 0.00 & 7.81 & 3.01 \end{pmatrix}$$

Step 3 (eliminating h_{23} and h_{32})

$$\mathbf{U}_3 = \begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 0.76 & -0.65 \\ 0.00 & 0.65 & 0.76 \end{pmatrix}$$

$$\mathbf{C}_3 = \mathbf{U}_3 \cdot \mathbf{C}_2 = \begin{pmatrix} 0.75 & 0.48 & -0.45 \\ -0.66 & 0.59 & 0.47 \\ 0.04 & 0.65 & 0.76 \end{pmatrix}$$

$$\boldsymbol{\epsilon}_3 = \mathbf{U}_3^T \boldsymbol{\epsilon}_2 \mathbf{U}_3 = \mathbf{C}_3^T \mathbf{h} \mathbf{C}_3 = \begin{pmatrix} -2.54 & 0.24 & -0.20 \\ 0.24 & 12.17 & 0.00 \\ -0.20 & 0.00 & -3.63 \end{pmatrix}$$

As we can see, as j increases $\boldsymbol{\epsilon}_j$ becomes incrementally closer to a diagonal matrix. The algorithm can be continued until all of the off-diagonal elements are smaller than some specified tolerance, at which point we have obtained the converged eigenvector matrix \mathbf{C} and diagonal eigenvalue matrix $\boldsymbol{\epsilon}$.

1.3 Jacobi Rotation Algorithm

In practical implementations, we typically make two changes to the algorithm described above. Firstly, for a more stable implementation, we do not actually calculate the rotation angle needed to eliminate matrix element h_{ij} , but instead use several trigonometric relations to calculate the required sine and cosine values directly. To do this, we first use Eq. (6) to define

$$\beta = \frac{c^2 - s^2}{2cs} = \frac{h_{ii} - h_{jj}}{2h_{ij}}. \quad (11)$$

We then use the following trigonometric identities to find an expression for $t = \tan \theta$ in terms of β :

$$t^2 + 2 \left(\frac{c^2 - s^2}{2cs} \right) \frac{s}{c} - 1 = \frac{s^2}{c^2} + \frac{c^2}{c^2} - \frac{s^2}{c^2} - 1 = 0 \quad (12)$$

$$t^2 + 2\beta t - 1 = 0 \quad (13)$$

$$t = \frac{\text{sgn}(\beta)}{|\beta| + \sqrt{\beta^2 + 1}}. \quad (14)$$

It is then straightforward to use more identities to find the sine and cosine values:

$$s^2 + c^2 = 1 = c^2(t^2 + 1), \quad c = \frac{1}{\sqrt{1 + t^2}}, \quad s = c \cdot t. \quad (15)$$

Secondly, we are often interested in the lowest or highest few eigenvalues of a matrix. Thus, after the completion of the Jacobi algorithm, we should sort the eigenvalues and associated eigenvectors in ascending order.

Finally, we must decide a suitable convergence condition at which to finish iterating. When diagonalizing a matrix, there are two common tests for convergence. Iterations are continued until either all of the off-diagonal elements of the diagonal matrix ϵ are below some specified threshold, or the changes in the diagonal elements of ϵ from one iteration to the next are below some specified threshold. This threshold must be chosen by the user; a common choice is 10^{-15} , which is approximately the limit of numerical precision of double-precision binary floating-point arithmetic, although for many applications a much larger threshold is sufficient.

With these modifications, the implementation of the Jacobi rotation algorithm to diagonalize an $n \times n$ matrix \mathbf{h} follows these steps:

1. Initialize $\mathbf{C} = \mathbf{1}$
2. Loop over the two indices of the off-diagonal elements: $i \in [2, n]; j \in [1, i - 1]$
 - (a) Calculate the values of s and c needed to eliminate the current off-diagonal element as follows:

$$\begin{aligned}\delta &= h_{ii} - h_{jj} \\ t &= \frac{2 \cdot \text{sgn}(\delta) \cdot h_{ij}}{|\delta| + \sqrt{|\delta|^2 + 4h_{ij}^2}} \\ c &= \frac{1}{\sqrt{1 + t^2}}, \quad s = c \cdot t\end{aligned}$$

- (b) Update \mathbf{C} , looping over all rows k , according to:

$$\begin{aligned}\tau &= cC_{kj} - sC_{ki} \\ C_{ki} &= sC_{kj} + cC_{ki} \\ C_{kj} &= \tau\end{aligned}$$

- (c) Update \mathbf{h} , looping over all rows $k \neq i, j$, according to:

$$\begin{aligned}\tau &= ch_{kj} - sh_{ki} \\ h_{ki} &= sh_{kj} + ch_{ki} \\ h_{kj} &= \tau \\ h_{ik} &= h_{ki}, \quad h_{jk} = h_{kj}\end{aligned}$$

- (d) Eliminate element h_{ij} :

$$\begin{aligned}\tau &= c^2h_{ii} + s^2h_{jj} + 2csh_{ij} \\ h_{jj} &= s^2h_{ii} + c^2h_{jj} - 2csh_{ij} \\ h_{ii} &= \tau \\ h_{ij} &= 0, \quad h_{ji} = 0\end{aligned}$$

3. Check for convergence. If not converged, go back to 2.
4. Sort the eigenvalues (and associated eigenvectors) in ascending order.

2 Numerical and Analytical Integration

In the Hartree-Fock algorithm, we will determine the molecular orbitals ϕ_i of a system, along with their eigenvalues. These orbitals are extremely complicated spatial functions, so to simplify the problem we express the molecular orbitals as a linear combination of atomic orbitals, ψ_μ :

$$\phi_i(r, \theta, \phi) = \sum_{\mu} \psi_{\mu}(r, \theta, \phi) c_{\mu i}. \quad (16)$$

In this chapter we will discuss a number of different possible representations of these atomic orbitals; these representations are called basis sets or basis functions. In order to solve the Hartree-Fock equations we will need to calculate various integrals using our chosen basis set. Therefore, we will also discuss the advantages and disadvantages of using different basis sets to evaluate these integrals.

2.1 Integrals for Hartree-Fock Calculations

Let $\psi_{\mu}(\beta, r, \theta, \phi)$ be our basis functions with some parameters β . The integrals (whose origin will be discussed later) needed to solve the Hartree-Fock equations are:

1. Overlap integrals:

$$S_{\mu\nu} = \int d\tau \psi_{\mu}(\beta_{\mu}, r, \theta, \phi) \psi_{\nu}(\beta_{\nu}, r, \theta, \phi) \quad (17)$$

2. Kinetic energy integrals:

$$t_{\mu\nu} = \int d\tau \psi_{\mu}(\beta_{\mu}, r, \theta, \phi) \left(\frac{-1}{2} \nabla^2 \right) \psi_{\nu}(\beta_{\nu}, r, \theta, \phi) \quad (18)$$

3. Nuclear attraction integrals:

$$v_{\mu\nu}^D = \int d\tau \psi_{\mu}(\beta_{\mu}, r, \theta, \phi) \left(\frac{-Z_D}{r_D} \right) \psi_{\nu}(\beta_{\nu}, r, \theta, \phi) \quad (19)$$

where Z_D is the charge of nucleus D and $r_D = r - \mathbf{r}_D$ with \mathbf{r}_D being the position of nucleus D .

4. Electron repulsion integrals:

$$G_{\mu\nu\mu'\nu'} = \int d\tau_1 \int d\tau_2 \psi_{\mu}(\beta_{\mu}, r_1, \theta_1, \phi_1) \psi_{\nu}(\beta_{\nu}, r_1, \theta_1, \phi_1) \left(\frac{1}{|r_1 - r_2|} \right) \\ \times \psi_{\mu'}(\beta_{\mu'}, r_2, \theta_2, \phi_2) \psi_{\nu'}(\beta_{\nu'}, r_2, \theta_2, \phi_2) \quad (20)$$

For simplicity, in all of the above we have contracted the variables of integration as

$$\int d\tau = \int r^2 dr \int \sin \theta d\theta \int d\phi. \quad (21)$$

2.2 Basis sets

So far we have not specified the form of the basis functions $\psi_\mu(\beta, r, \theta, \phi)$ used in Section 2.1. One natural choice are functions analogous to the known solutions of the hydrogen atom, which are of the form

$$\psi_\mu(\alpha, n, l, m, r, \theta, \phi) = R_{nl}(r, \alpha) Y_{lm}(\theta, \phi), \quad (22)$$

where n , l and m are the well known atomic quantum numbers, Y_{lm} are spherical harmonics, and the radial function R_{nl} is given by

$$R_{nl}(r, \alpha) = N (2\alpha r)^l [L_{n-l-1}^{2l+1}(2\alpha r)] e^{-\alpha r}, \quad (23)$$

where $L_{n-l-1}^{2l+1}(2\alpha r)$ are the generalized Laguerre polynomials, and N is a normalization factor. For atoms with only one electron, α is given exactly by $Z\mu/(nm_e a_0)$, where Z is the nuclear charge, μ is the reduced mass, m_e is the mass of an electron, and a_0 is the Bohr radius. However, for any atom with more than one electron these orbitals are only approximate solutions, with the nuclear charge Z replaced with a screened effective charge parameter Z_{eff} , typically calculated using the variational principle.

Unfortunately, it is not possible to find analytical solutions to the one- and two-electron integrals in Section 2.1 using these hydrogenic basis functions. Instead, the integrals must be evaluated numerically. This is computationally expensive, especially given the complicated form of the radial function R_{nl} . Nevertheless, some modern computational codes do use hydrogenic basis functions and numerically evaluate the one- and two-electron integrals; this will be discussed further in Section 2.3.

Another possible choice of basis functions are those which resemble hydrogenic orbitals, but are markedly simpler and, importantly, do not contain nodes:

$$\psi_\mu(\alpha, n, l, m, r, \theta, \phi) = N r^{n-1} e^{-\alpha r} Y_{lm}(\theta, \phi). \quad (24)$$

These functions are known as Slater-type orbitals (STOs). Using these basis functions does allow analytical solutions to the one- and two-electron integrals, but these analytical solutions are still relatively expensive to evaluate numerically. STOs are rarely used in practice in modern computational chemistry, but were an important step in the development of basis sets and are often referred to in chemistry textbooks, and are still used as reference points for other basis sets.

Instead, a common approach is to use basis functions whose integrals are much easier to solve, and then to use enough of these functions to approximately reproduce STOs or hydrogenic orbitals. To this end, Gaussian functions (or Gaussian-type orbitals, GTOs) of the form

$$\psi_i(\alpha_i, l, m, r, \theta, \phi) = r^l e^{-\alpha_i r^2} Y_{lm}(\theta, \phi) \quad (25)$$

are typically used. A single GTO differs from the hydrogenic orbital in two ways, which are illustrated in Figure 1. First, it misses the cusp at the origin and instead flattens out close to $r = 0$. Second, at larger distances it drops off much more quickly, and thus a single GTO does not extend as far as an hydrogenic orbital.

In order to mitigate these problems, a single basis function ψ_μ is defined as a linear

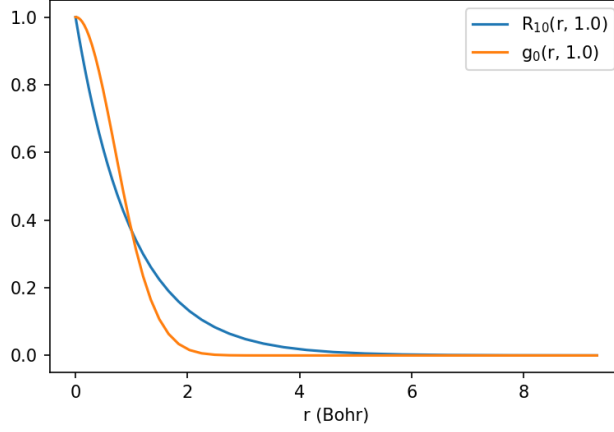


Figure 1: Comparison between the radial parts of a hydrogenic orbital and a single Gaussian function.

combination of M Gaussian functions, weighted by coefficients λ :

$$\begin{aligned}\psi_\mu(\{\lambda_i\}, \{\alpha_i\}, l, m, r, \theta, \phi) &= \sum_{i=1}^M \lambda_i \psi_i(\alpha_i, l, m, r, \theta, \phi) \\ &= r^l Y_{lm}(\theta, \phi) \sum_{i=1}^M \lambda_i e^{-\alpha_i r^2},\end{aligned}\quad (26)$$

which is also known as contraction. By combining Gaussian functions with very different exponents, this contraction scheme can alleviate the problems which arise from using a single Gaussian function, as shown in see Figure 2.

When using these contracted Gaussian basis functions, the integrals in Section 2.1 have analytical solutions which are very efficient to evaluate. These solutions will be discussed in more detail in Section 2.4, and are listed in the Appendix. Typical nomenclature of basis sets will tell how many Gaussian functions are contracted and which types of basis functions are included.

2.3 Numerical Integration of Hydrogenic Basis Sets

As mentioned in the previous section, hydrogenic functions are very good representations for the atomic orbitals (accurate and efficient), but they need to be integrated numerically. In order to understand the effort involved in numerical integration, and how one can overcome some of the difficulties which arise, consider the radial part of two hydrogenic functions centered at 0 and calculate a hypothetical overlap, namely

$$s_{\mu\nu} = \int_0^\infty dr \psi_\mu(r) \psi_\nu(r) = \int_0^\infty dr R_{10}(r, 1.0) R_{10}(r, 0.5), \quad (27)$$

where R is given by Eq. (23). The two functions and the integrand of the equation above is shown in Fig. 3.

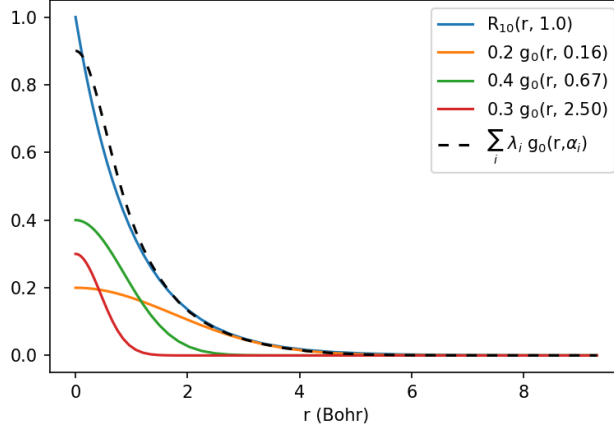


Figure 2: Example of a contraction of three Gaussian functions (orange, green, red) to best match a single hydrogenic orbital

In general, performing a numerical integration involves evaluating

$$\int_0^{r_{max}} f(r) dr \approx \sum_{j=0}^{N_{max}} w_j f(r_j) \quad (28)$$

where w_j is a weight related to a given integration point r_j . One simple example of such an integration scheme is the well-known trapezium rule, for which the weights are simply given by $w_j = \Delta r_j = r_j - r_{j-1}$ for all points $0 < j < N$, and $w_j = \Delta r_j/2$ for $j = 0, N$. Many other more complicated integration schemes are also widely used, such as Simpson's rule and Gaussian quadrature.

Using a uniform grid to perform the simple integral $s_{\mu\nu}$ numerically by the simplest method, i.e. the “left-hand rule”, such that the integration points are defined by $r_j = j(r_{max}/N_{max})$, $j = 1, \dots, N_{max}$, and $w_j = r_{max}/N_{max}$, requires around 7000 points to achieve a 0.1% absolute error in the evaluation of Eq. (27).

Instead, much better grid point positions and weights can be defined for this problem. For example, a radial integration grid proposed by B. Delley [*c.f.* J. Chem. Phys. 92, 508 (1990)] to integrate these types of function is given by

$$r_j = r_{max} \frac{\ln(1 - (j/(N_{max} + 1))^2)}{\ln(1 - (N_{max}/(N_{max} + 1))^2)}. \quad (29)$$

The integration weights can be obtained by considering j to be a continuous variable and obtaining the following result in one dimension,

$$w_j = \frac{dr}{dj} = - \frac{2jr_{max}}{((N_{max} + 1)^2 - j^2) \ln(1 - (N_{max}/(N_{max} + 1))^2)}. \quad (30)$$

The comparison of these two grids (20 points each) is shown in Figure 4. It is already apparent that the grids in Eq. (29) are more densely distributed where the change in the

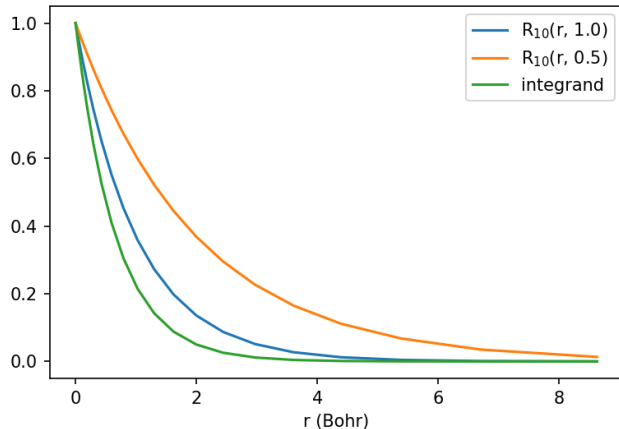


Figure 3: Radial parts of hydrogenic functions.

function is steeper, at smaller r . Indeed, the same error of around 0.1% in the evaluation of Eq. (27) can be achieved with only 30 points! While this comparison is slightly unfair, because it is quite easy to improve the quality of the numerical integration even while using uniform grids, it gives a good idea of how much computational power can be saved by using tailored grids. In real calculations, the angular part of the integration over spherical harmonics can also be performed on tailored grids.

Some electronic structure codes choose to follow this route and use numeric atom-centered orbitals based on numerically tabulated hydrogenic functions as their basis sets. Because the numerical integration can be made quite efficient, and parallelization of these operations over many computational cores is easy to achieve, the advantage of having small and accurate basis sets can win over being able to perform such integrals analytically.

2.4 Analytical Integration of Gaussian Basis Sets

In Section 2.2, we showed how a single basis function could be constructed from many Gaussian functions, such that its radial part resembled a hydrogenic or Slater-type orbital. As a result, the number of individual functions used in a Gaussian basis set is typically many times larger than the number of functions used in a comparable hydrogenic basis set. Nevertheless, the integrals defined in Section 2.1 may often be evaluated more quickly when using Gaussian basis sets, since they can be solved analytically. In this section, we describe how the integrals of Gaussian basis functions may be simplified, and derive the analytical solution for the overlap integral of two GTOs.

2.4.1 Sneaky Tricks

There are a few mathematical tricks which we may exploit when evaluating the one- and two-electron integrals in Section 2.1 when using GTO basis functions. Firstly, within the contraction scheme we can break down the integrals of the basis functions to integrals of individual Gaussian functions. This will be demonstrated for the overlap integrals, but holds

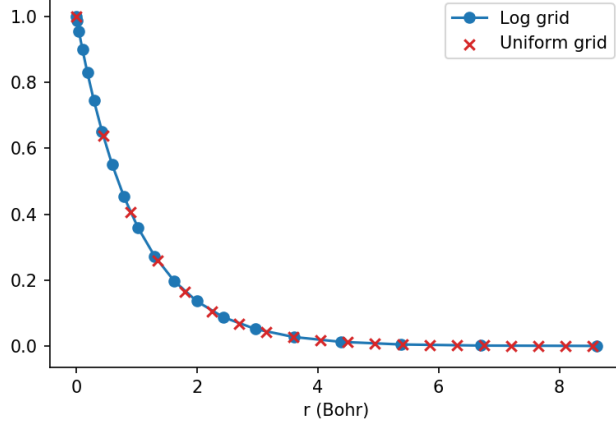


Figure 4: Comparison between different grids with 20 points in the given interval for a hydrogenic function.

true for all other integrals. Considering the overlap integral of basis functions μ and ν , and inserting the definition of a contracted GTO from Eq. (26), gives:

$$\begin{aligned}
S_{\mu\nu} &= \int d\tau \psi_{\mu}(\beta_{\mu}, r, \theta, \phi) \psi_{\nu}(\beta_{\nu}, r, \theta, \phi) \\
&= \int d\tau \left(\sum_{i=1}^{M_{\mu}} \lambda_i^{\mu} \psi_i^{\mu}(\beta_i^{\mu}, r, \theta, \phi) \right) \left(\sum_{j=1}^{M_{\nu}} \lambda_j^{\nu} \psi_j^{\nu}(\beta_j^{\nu}, r, \theta, \phi) \right) \\
&= \sum_{i=1}^{M_{\mu}} \lambda_i^{\mu} \sum_{j=1}^{M_{\nu}} \lambda_j^{\nu} \int d\tau \psi_i^{\mu}(\beta_i^{\mu}, r, \theta, \phi) \psi_j^{\nu}(\beta_j^{\nu}, r, \theta, \phi) \\
&= \sum_{i=1}^{M_{\mu}} \sum_{j=1}^{M_{\nu}} \lambda_i^{\mu} \lambda_j^{\nu} \tilde{S}_{i,j}^{\mu,\nu}.
\end{aligned} \tag{31}$$

We can see that $\tilde{S}_{i,j}^{\mu,\nu}$ is now the overlap of two individual Gaussian functions: ψ_i^{μ} , the i th function in the contraction which forms the basis function μ , and ψ_j^{ν} , the j th function in the contraction which forms the basis function ν .

In all of the above we have assumed that every basis function is centred on a single atom found at the origin. Of course, when considering molecules this assumption is no longer valid. Instead, we must shift the centre of the basis function μ to the position of the atom A with which it is associated. That is, the position of the electron is defined relative to the position of the nucleus of atom A , \mathbf{r}_A : $\mathbf{r} \rightarrow \mathbf{r} - \mathbf{r}_A$. The resulting modifications to the basis functions defined above are relatively straightforward. In the radial part of the function, any appearance of r is replaced with $r_A = |\mathbf{r} - \mathbf{r}_A|$. Similarly, the real spherical harmonics $Y_{lm}(\theta, \phi)$ can be written as functions of the Cartesian coordinates; these are simply replaced with $x - x_A$, $y - y_A$ and $z - z_A$.

(l, m)	(a, b, c)
(0,0)	(0,0,0)
(1,-1)	(0,1,0)
(1,0)	(0,0,1)
(1,1)	(1,0,0)
(2,-2)	(1,1,0)
(2,-1)	(0,1,1)
(2,0)	$\frac{1}{2}(2 \times (0, 0, 2) - (2, 0, 0) - (0, 2, 0))$
(2,1)	(1,0,1)
(2,-2)	$\sqrt{\frac{3}{4}}((2, 0, 0) - (0, 2, 0))$

Table 1: A spherical GTO with quantum numbers (l, m) can be written as a linear combination of Cartesian GTOs with quantum numbers (a, b, c)

At this point it is helpful to define the Cartesian GTO:

$$\psi_i^C(\alpha_i, a, b, c, x, y, z) = N_i(x - x_A)^a(y - y_A)^b(z - z_A)^c e^{-\alpha_i r_A^2}. \quad (32)$$

These functions can be straightforwardly related to the GTOs defined in Eq. (25), which are known as spherical GTOs: each spherical GTO with quantum numbers l, m can be written as a linear combination of Cartesian GTOs whose quantum numbers satisfy $a + b + c = l$. Table 1 lists the relationship between selected spherical and Cartesian GTOs. As we will see shortly, introducing these Cartesian GTOs will make evaluating the integrals in Section 2.1 significantly easier when the basis functions are not centred at the origin.

The final and key advantage of using GTOs as basis functions comes from the *Gaussian product rule*. This states that the product of any two Gaussian functions centred at different positions is proportional to a single Gaussian centred at a weighted midpoint. As an example, consider the product of two exponential functions with different exponents centred on points \mathbf{r}_A and \mathbf{r}_B :

$$e^{-\alpha r_A^2} e^{-\beta r_B^2} = e^{-\frac{\alpha\beta}{\alpha+\beta} r_{AB}^2} e^{-(\alpha+\beta) r_P^2}, \quad (33)$$

where $r_{AB} = |\mathbf{r}_A - \mathbf{r}_B|$, $r_P = |\mathbf{r} - \mathbf{r}_P|$ and $\mathbf{r}_P = \frac{\alpha\mathbf{r}_A + \beta\mathbf{r}_B}{\alpha+\beta}$. A proof of Eq. (33) is left as an exercise.

2.4.2 Evaluating the Overlap Integrals

Combining all of these tricks and observations, we are able to arrive at analytical (although not simple!) expressions for the integrals defined in Section 2.1. Here, we will work through how to obtain an expression for the overlap integral $S_{\mu\nu}$; the expressions for the other integrals will be listed in Appendix A.

From Eq. (31), we see that in order to determine the total overlap integral $S_{\mu\nu}$ we must find an expression for the overlap of two Gaussians, $\tilde{S}_{i,j}^{\mu,\nu}$. This is defined as

$$\tilde{S}_{i,j}^{\mu,\nu} = \int d\tau \psi_i^\mu(\beta_i^\mu, r, \theta, \phi) \psi_j^\nu(\beta_j^\nu, r, \theta, \phi). \quad (34)$$

The first step is to convert from spherical GTOs to Cartesian GTOs. To do this, we make the substitution:

$$\psi_i^\mu(\alpha_i^\mu, l, m, r, \theta, \phi) = \sum_k \lambda_k^C \psi_{i,k}^{\mu,C}(\alpha_i^\mu, a_k, b_k, c_k, x, y, z), \quad (35)$$

where the set of quantum numbers $\{a_k, b_k, c_k\}$ are determined by l and m as in Table 1, and λ_k^C is the appropriate weight for the k th Cartesian GTO, again shown in Table 1. Choosing Cartesian variables of integration, $\int d\boldsymbol{\tau} = \int \int \int dx dy dz$:

$$\begin{aligned}\tilde{S}_{i,j}^{\mu,\nu} &= \sum_{k,l} \lambda_k^C \lambda_l^C \int d\boldsymbol{\tau} \psi_{i,k}^{\mu,C}(\alpha_i^\mu, a_k, b_k, c_k, x, y, z) \psi_{j,l}^{\nu,C}(\alpha_j^\nu, a_l, b_l, c_l, x, y, z) \\ &= \sum_{k,l} \int d\boldsymbol{\tau} N_i^k (x - x_A)^{a_k} (y - y_A)^{b_k} (z - z_A)^{c_k} e^{-\alpha_i^\mu r_A^2} \\ &\quad \times N_j^l (x - x_B)^{a_l} (y - y_B)^{b_l} (z - z_B)^{c_l} e^{-\alpha_j^\nu r_B^2}.\end{aligned}\quad (36)$$

Here we have introduced the weighted normalization constant N_i^k , which is the product of the normalization constant for the Cartesian GTO and λ_k^C :

$$N_i^k = \lambda_k^C \left(\frac{2\alpha_i}{\pi} \right)^{3/4} \left(\frac{(8\alpha_i)^{a_k+b_k+c_k} a_k! b_k! c_k!}{(2a_k)!(2b_k)!(2c_k)!} \right)^{1/2}. \quad (37)$$

We can now use the Gaussian product rule (Eq. (33)) to combine the two exponential functions into a single Gaussian multiplied by a constant prefactor:

$$\begin{aligned}\tilde{S}_{i,j}^{\mu,\nu} &= \sum_{k,l} \int d\boldsymbol{\tau} N_i^k (x - x_A)^{a_k} (y - y_A)^{b_k} (z - z_A)^{c_k} e^{-\frac{\alpha_i^\mu \alpha_j^\nu}{\alpha_i^\mu + \alpha_j^\nu} r_{AB}^2} e^{-(\alpha_i^\mu + \alpha_j^\nu) r_P^2} \\ &\quad \times N_j^l (x - x_B)^{a_l} (y - y_B)^{b_l} (z - z_B)^{c_l},\end{aligned}\quad (38)$$

where as before $r_{AB} = |\mathbf{r}_A - \mathbf{r}_B|$, $r_P = |\mathbf{r} - \mathbf{r}_P|$ and $\mathbf{r}_P = \frac{\alpha_i^\mu \mathbf{r}_A + \alpha_j^\nu \mathbf{r}_B}{\alpha_i^\mu + \alpha_j^\nu}$. Since we can write $r_P^2 = (x - x_P)^2 + (y - y_P)^2 + (z - z_P)^2$, the integral is now separable:

$$\begin{aligned}\tilde{S}_{i,j}^{\mu,\nu} &= e^{-\frac{\alpha_i^\mu \alpha_j^\nu}{\alpha_i^\mu + \alpha_j^\nu} r_{AB}^2} \sum_{k,l} N_i^k N_j^l \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} \\ \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} &= \int dx (x - x_A)^{a_k} (x - x_B)^{a_l} e^{-(\alpha_i^\mu + \alpha_j^\nu)(x - x_P)^2} \\ \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} &= \int dy (y - y_A)^{b_k} (y - y_B)^{b_l} e^{-(\alpha_i^\mu + \alpha_j^\nu)(y - y_P)^2} \\ \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} &= \int dz (z - z_A)^{c_k} (z - z_B)^{c_l} e^{-(\alpha_i^\mu + \alpha_j^\nu)(z - z_P)^2}.\end{aligned}\quad (39)$$

We can then use binomial expansions to further simplify the remaining integrals. For example:

$$\begin{aligned}(x - x_A)^{a_k} (x - x_B)^{a_l} &= [(x - x_P) + (x_P - x_A)]^{a_k} [(x - x_P) + (x_P - x_B)]^{a_l} \\ &= \sum_{m=0}^{a_k} \sum_{n=0}^{a_l} \binom{a_k}{m} \binom{a_l}{n} (x_P - x_A)^{a_k-m} (x_P - x_B)^{a_l-n} (x - x_P)^{m+n}.\end{aligned}\quad (40)$$

Inserting this into the definition of $\tilde{X}_{i,j,a_k,a_l}^{\mu,\nu}$, we find

$$\begin{aligned} \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} &= \sum_{m=0}^{a_k} \sum_{n=0}^{a_l} \binom{a_k}{m} \binom{a_l}{n} (x_P - x_A)^{a_k-m} (x_P - x_B)^{a_l-n} \\ &\times \int dx (x - x_P)^{m+n} e^{-(\alpha_i^\mu + \alpha_j^\nu)(x-x_P)^2}. \end{aligned} \quad (41)$$

The remaining integral now has a known closed form:

$$\int_{-\infty}^{\infty} dx (x - x_P)^{m+n} e^{-(\alpha_i^\mu + \alpha_j^\nu)(x-x_P)^2} = \begin{cases} 0, & \text{if } m+n \text{ is odd} \\ \frac{\sqrt{2\pi}(m+n-1)!!}{[2(\alpha_i^\mu + \alpha_j^\nu)]^{(m+n+1)/2}}, & \text{if } m+n \text{ is even} \end{cases}, \quad (42)$$

where !! indicates a double factorial. The solutions to integrals $\tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu}$ and $\tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu}$ have an entirely analogous closed form to that which we have found for $\tilde{X}_{i,j,a_k,a_l}^{\mu,\nu}$. Therefore, by combining Eqs. (31), (35), (39), (41) and (42) we have an analytical result for the overlap integral $S_{\mu\nu}$.

3 Hartree-Fock Implementation

In this section we will review the Hartree-Fock method of calculating the molecular orbitals and total energy of a many-electron system.

3.1 Definitions

The overall non-relativistic Hamiltonian for a molecule is:

$$\hat{H} = \hat{H}_{nuc} + \hat{H}_{el} \quad (43)$$

$$\hat{H}_{nuc} = -\sum_A \frac{1}{2M_A} \nabla_A^2 + \frac{1}{2} \sum_{A \neq B} \frac{Z_A Z_B}{r_{AB}} \quad (44)$$

$$\hat{H}_{el} = -\sum_i \frac{1}{2} \nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{iA}} + \sum_{i \neq j} \frac{1}{r_{ij}}. \quad (45)$$

Here Z_A and M_A are the charge and mass of nucleus A respectively, r_{AB} is the distance between nucleus A and B , r_{iA} is the distance between electron i and nucleus A , r_{ij} is the distance between electrons i and j , and we have used atomic units, setting various physical constants to 1. Following the Born-Oppenheimer approximation, where we assume the positions of all nuclei are fixed, we obtain the Schrödinger equation for the electrons:

$$\hat{H}_{el} \chi_k = \epsilon_k \chi_k. \quad (46)$$

In Hartree-Fock we treat the interaction of each electron with all other electrons at the mean field level. As a result the final Hartree-Fock equations read

$$\hat{f} \chi_k = \epsilon_k \chi_k, \quad k = 1, 2, \dots, N, \quad (47)$$

where N is the number of electrons and the Fock operator is defined as

$$\hat{f} = \hat{h} + \sum_j (\hat{J}_j - \hat{K}_j) \quad (48)$$

$$\hat{h} \chi_k(r_1) = -\sum_i \frac{1}{2} \nabla_i^2 \chi_k(r_1) - \sum_{i,A} \frac{Z_A}{r_{iA}} \chi_k(r_1) \quad (49)$$

$$\hat{J}_j \chi_k(r_1) = \chi_k(r_1) \int dr_2 \chi_j^*(r_2) \frac{1}{r_{12}} \chi_j(r_2) \quad (50)$$

$$\hat{K}_j \chi_k(r_1) = \chi_j(r_1) \int dr_2 \chi_j^*(r_2) \frac{1}{r_{12}} \chi_k(r_2). \quad (51)$$

\hat{h} is called as the single-particle operator, \hat{J}_j as the Coulomb operator and \hat{K}_j as the exchange operator. For many-electron systems, the Fock operator \hat{f} can be thought of as an approximation to \hat{H}_{el} .

3.2 Closed-Shell Restricted Hartree-Fock

Now we will take steps towards the practical solution of the Hartree-Fock equations. For atoms it is possible to solve the complicated, non-linear integro-differential equation directly, but for molecules this is very tedious and not practical. At the end of the 1940s both

Roothaan in Chicago and Hall in Cambridge found a suitable and practical way, which we will discuss now.

First, we have to consider the spin of the electron, which we have neglected so far. Due to the Pauli exclusion principle, two fermions have to differ in at least one quantum number. Electrons have a spin of $1/2$ and thus two different spin projections are possible, $m_s = \pm 1/2$. As the spin does not show up in the non-relativistic Hamiltonian we are considering, both projections give the same contribution to the energy. Now we identify the single-particle functions χ_k as so-called spin orbitals, which can be represented as a product of a space function $\phi_k(\mathbf{r})$ and a spin function $|\alpha_k\rangle$ or $|\beta_k\rangle$:

$$\chi_k = \phi_k(\mathbf{r}) |\sigma_k\rangle; \quad \sigma = \alpha, \beta. \quad (52)$$

The scalar product of two single-particle functions is

$$\langle \chi_i | \chi_j \rangle = \int d\mathbf{r} \phi_i^*(\mathbf{r}) \phi_j(\mathbf{r}) \langle \sigma_i | \sigma_j \rangle, \quad (53)$$

with $\langle \sigma_i | \sigma_j \rangle = \delta_{\sigma_i \sigma_j}$.

We can now insert this definition of the single particle functions χ_k into the Hartree-Fock equations:

$$\hat{f}(1) \phi_k(\mathbf{r}_1) |\sigma_k(1)\rangle = \epsilon_k \phi_k(\mathbf{r}_1) |\sigma_k(1)\rangle. \quad (54)$$

Multiplying this from the left with a spin function $\langle \sigma_k(1) |$:

$$\begin{aligned} \langle \sigma_k(1) | \hat{f}(1) | \sigma_k(1) \rangle \phi_k(\mathbf{r}_1) &= \epsilon_k \phi_k(\mathbf{r}_1) \langle \sigma_k(1) | \sigma_k(1) \rangle \\ &= \langle \sigma_k(1) | \hat{h}(1) | \sigma_k(1) \rangle \phi_k(\mathbf{r}_1) + \sum_j \langle \sigma_k(1) | \hat{J}_j(1) | \sigma_k(1) \rangle \phi_k(\mathbf{r}_1) \\ &\quad - \langle \sigma_k(1) | \phi_j(\mathbf{r}_1) | \sigma_j(1) \rangle \int d\mathbf{r}_2 \langle \sigma_j(2) | \phi_j^*(\mathbf{r}_2) \frac{1}{r_{12}} | \sigma_k(2) \rangle \phi_k(\mathbf{r}_2) \\ &= \hat{h}(1) \phi_k(\mathbf{r}_1) + \sum_j \hat{J}_j(1) \phi_k(\mathbf{r}_1) \\ &\quad - \langle \sigma_k(1) | \sigma_j(1) \rangle \langle \sigma_j(2) | \sigma_k(2) \rangle \hat{K}_j(1) \phi_k(\mathbf{r}_1) \\ &= \left[\hat{h}(1) + \sum_j \left(\hat{J}_j(1) - \delta_{\sigma_j \sigma_k} \hat{K}_j(1) \right) \right] \phi_k(\mathbf{r}_1) = \epsilon_k \phi_k(\mathbf{r}_1). \end{aligned} \quad (55)$$

From Eq. (55) we see that the exchange operator only acts if χ_j and χ_k have the same spin function. For a system with an even number of electrons (like most molecules) this will result in a ground state where the lowest energy $N/2$ space orbitals ϕ_k are occupied with $|\alpha\rangle$ and $|\beta\rangle$ spin functions, respectively. We thus only sum over $N/2$ space orbitals in the Fock operator and multiply $\hat{h}(1)$ and the Coulomb operator by two. This results in the closed-shell restricted Hartree-Fock equations:

$$\left[2\hat{h}(1) + \sum_j^{N/2} \left(2\hat{J}_j(1) - \hat{K}_j(1) \right) \right] \phi_k(\mathbf{r}_1) = \epsilon_k \phi_k(\mathbf{r}_1). \quad (56)$$

Within the Hartree-Fock method, the following notation is commonly used for the one-

and two- electron integrals over the space functions ϕ_i :

$$\int d\mathbf{r}_1 \phi_i^*(\mathbf{r}_1) \hat{h}(1) \phi_j(\mathbf{r}_1) = (i|\hat{h}|j) \quad (57)$$

$$\int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_i^*(\mathbf{r}_1) \phi_j(\mathbf{r}_1) \frac{1}{r_{12}} \phi_k^*(\mathbf{r}_2) \phi_l(\mathbf{r}_2) = (ij|kl) \quad (58)$$

With these definitions, the restricted Hartree-Fock energy is:

$$E_{\text{HF}} = 2 \sum_i^{N/2} (i|\hat{h}|i) + \sum_{i<j}^{N/2} 2(i|j|j) - (ij|ji) \quad (59)$$

3.3 Linear Combination of Atomic Orbitals

In the last section, we solved the spin-part of the Hartree-Fock problem (at least for closed-shell cases). In this section, we will discuss a practical solution for the space orbitals. The molecular orbitals we are interested in are complicated functions in space. Thus, we will represent them as a linear combination of simpler functions, called atomic orbitals, located at each atom of the molecule. We have already discussed some possible choices for representing these atomic orbitals in Section 2.2. The ansatz is:

$$\phi_i = \sum_{\mu} \psi_{\mu} c_{\mu i}. \quad (60)$$

Every molecular orbital ϕ_i is represented by the atomic orbitals ψ_{μ} with expansion coefficients $c_{\mu i}$. However, the atomic orbitals centered at different atoms are not orthogonal, while the molecular orbitals are required to be orthogonal. We find:

$$\begin{aligned} \langle \phi_i | \phi_j \rangle &= \left\langle \sum_{\mu} \psi_{\mu} c_{\mu i} \left| \sum_{\nu} \psi_{\nu} c_{\nu j} \right. \right\rangle \\ &= \sum_{\mu, \nu} c_{\mu i}^* c_{\nu j} \langle \psi_{\mu} | \psi_{\nu} \rangle = \sum_{\mu, \nu} c_{\mu i}^* c_{\nu j} S_{\mu \nu}, \end{aligned} \quad (61)$$

where $S_{\mu \nu}$ is exactly the overlap integral defined in Section 2.1. The matrix of these integrals \mathbf{S} is hermitian and positive semi-definite, and we can write the scalar product in matrix form as

$$\langle \phi_i | \phi_j \rangle = \mathbf{c}_i^{\dagger} \mathbf{S} \mathbf{c}_j, \quad (62)$$

where \mathbf{c}_i is a vector of the expansion coefficients for the i -th molecular orbital.

We can now insert the ansatz into the restricted Hartree-Fock equations and project onto the atomic orbital $|\psi_{\mu}\rangle$ to get

$$\begin{aligned} \hat{f}(1) \phi_i &= \epsilon_i \phi_i \\ \hat{f}(1) \sum_{\nu} \psi_{\nu} c_{\nu i} &= \epsilon_i \sum_{\nu} \psi_{\nu} c_{\nu i} \\ \sum_{\nu} \langle \psi_{\mu} | \hat{f}(1) | \psi_{\nu} \rangle c_{\nu i} &= \epsilon_i \sum_{\nu} \langle \psi_{\mu} | \psi_{\nu} \rangle c_{\nu i} \\ \sum_{\nu} F_{\mu \nu} c_{\nu i} &= \epsilon_i \sum_{\nu} S_{\mu \nu} c_{\nu i}. \end{aligned} \quad (63)$$

In matrix notation this can be written as

$$\mathbf{F}\mathbf{c}_i = \epsilon_i \mathbf{S}\mathbf{c}_i. \quad (64)$$

Since this holds true for every molecular orbital ϕ_i , we may define \mathbf{C} as the coefficient matrix for complete set of molecular orbitals, in which the i th column of the matrix contains the coefficients \mathbf{c}_i , and write the entire system of equations as

$$\mathbf{F}\mathbf{C} = \epsilon \mathbf{S}\mathbf{C}. \quad (65)$$

These are the Roothaan-Hall equations. \mathbf{F} is the Fock matrix in the atomic orbital basis, and ϵ is a diagonal matrix which contains the single particle energies (molecular orbital energies). In this basis, the Fock matrix is constructed using the kinetic energy, nuclear attraction and electron repulsion integrals defined in Section 2.1:

$$F_{\mu\nu} = t_{\mu\nu} + v_{\mu\nu} + \sum_{\mu'\nu'} \rho_{\mu'\nu'} (2G_{\mu\nu\mu'\nu'} - G_{\mu\nu'\mu'\nu}), \quad (66)$$

where $\rho_{\mu'\nu'} = \sum_i^{N/2} c_{\mu'i}^* c_{\nu'i}$ is the density matrix in the atomic orbital basis.

The Fock matrix is non-linear and depends on its own solution, and thus the Roothaan-Hall equations are a pseudo eigenvalue problem. Furthermore, as \mathbf{S} is not diagonal the pseudo eigenvalue problem is actually a generalized pseudo eigenvalue problem. Thus, we cannot solve this equation by direct diagonalization of the Fock matrix. We have to orthogonalize the basis first. For that we need to find a non-unitary transformation matrix \mathbf{X} with

$$\mathbf{X}^\dagger \mathbf{S} \mathbf{X} = \mathbf{I}. \quad (67)$$

To obtain \mathbf{X} we diagonalize \mathbf{S} as

$$\mathbf{U}^\dagger \mathbf{S} \mathbf{U} = \mathbf{s}, \quad (68)$$

where \mathbf{s} is a diagonal matrix. Since \mathbf{S} is positive semi-definite, we can choose $\mathbf{X} = \mathbf{U}\mathbf{s}^{-1/2}$ and obtain

$$\mathbf{X}^\dagger \mathbf{S} \mathbf{X} = \mathbf{s}^{-1/2} \mathbf{U}^\dagger \mathbf{S} \mathbf{U} \mathbf{s}^{-1/2} = \mathbf{s}^{-1/2} \mathbf{s} \mathbf{s}^{-1/2} = \mathbf{I}. \quad (69)$$

This is the so-called canonical diagonalization. One could also use different approaches. For the full Hartree-Fock equation we now obtain

$$\mathbf{F}\mathbf{X}\mathbf{X}^{-1}\mathbf{C} = \epsilon \mathbf{S}\mathbf{X}\mathbf{X}^{-1}\mathbf{C}. \quad (70)$$

The transformed coefficient matrix is denoted as $\mathbf{C}' = \mathbf{X}^{-1}\mathbf{C}$, and we multiply both sides with \mathbf{X}^\dagger from the left to get

$$\mathbf{X}^\dagger \mathbf{F} \mathbf{X} \mathbf{C}' = \epsilon \mathbf{X}^\dagger \mathbf{S} \mathbf{X} \mathbf{C}'. \quad (71)$$

Now we denote the transformed Fock matrix as $\mathbf{F}' = \mathbf{X}^\dagger \mathbf{F} \mathbf{X}$ to obtain a typical eigenvalue equation:

$$\mathbf{F}' \mathbf{C}' = \epsilon \mathbf{C}'. \quad (72)$$

We can use the solutions \mathbf{C}' to solve the generalized pseudo eigenvalue problem iteratively.

3.4 Self-Consistent Field Approach

We now have everything at hand to solve the Roothaan-Hall equations numerically. As the equations are non-linear, we have to solve them iteratively until the Fock operator is self-consistent. The procedure is as follows:

1. Get molecular geometry and atomic orbital basis set.
2. Calculate nuclear repulsion energy $E_{nr} = \frac{1}{2} \sum_{A \neq B} Z_A Z_B / R_{AB}$.
3. Calculate all integrals defined in Section 2.1: $S_{\mu\nu}$, $t_{\mu\nu}$, $v_{\mu\nu}$, and $G_{\mu\nu\mu'\nu'}$.
4. Diagonalize \mathbf{S} and obtain $\mathbf{X} = \mathbf{U}\mathbf{S}^{-1/2}$.
5. Obtain core Hamiltonian matrix $\mathbf{h} = \mathbf{t} + \mathbf{v}$.
6. Initialise the first coefficient matrix $\mathbf{C} = 0$.
7. Calculate the density matrix

$$\rho_{\mu\nu} = \sum_k c_{\mu k} c_{\nu k} n_k, \quad (73)$$

where $n_k = 1$ if the orbital k is occupied and 0 otherwise.

8. Iterate the following steps until convergence:

- (a) Build the Fock matrix

$$\Gamma_{\mu\nu} = \sum_{\mu'\nu'} \rho_{\mu'\nu'} (2G_{\mu\nu\mu'\nu'} - G_{\mu\nu'\mu'\nu}); \quad \mathbf{F} = \mathbf{h} + \mathbf{\Gamma}. \quad (74)$$

- (b) Transform Fock matrix $\mathbf{F}' = \mathbf{X}^\dagger \mathbf{F} \mathbf{X}$.
 - (c) Diagonalize \mathbf{F}' to obtain \mathbf{C}' and ϵ .
 - (d) Backtransform coefficients $\mathbf{C} = \mathbf{X} \mathbf{C}'$.
 - (e) Build the new density matrix ρ .
 - (f) Check convergence: did \mathbf{C} change significantly?
9. Calculate energy $E_{\text{HF}} = \text{tr}(\rho \mathbf{F}) + \text{tr}(\rho \mathbf{h}) + E_{nr}$.

A Integrals of GTOs

In this appendix, we summarise the analytic expressions obtained when using contracted Gaussian basis functions to evaluate the four integrals introduced in Section 2.1. The scalar distance between two nuclei A and B is written as r_{AB} , and the weighted midpoint between these nuclei P for Gaussian functions i and j is defined using Eq. (33):

$$\mathbf{r}_P = \frac{\alpha_i \mathbf{r}_A + \alpha_j \mathbf{r}_B}{\alpha_i + \alpha_j}. \quad (75)$$

The weighted normalization coefficients N_i^k are given by

$$N_i^k = \lambda_k^C \left(\frac{2\alpha_i}{\pi} \right)^{3/4} \left(\frac{(8\alpha_i)^{a_k+b_k+c_k} a_k! b_k! c_k!}{(2a_k)!(2b_k)!(2c_k)!} \right)^{1/2}, \quad (76)$$

where λ_k^C is the appropriate weight to ensure that the sum over k Cartesian GTOs reproduces a spherical GTO, as outlined in Table 1.

A.1 Overlap Integrals

$$\begin{aligned} S_{\mu\nu} &= \sum_{i=1}^{M_\mu} \sum_{j=1}^{M_\nu} \lambda_i^\mu \lambda_j^\nu \tilde{S}_{i,j}^{\mu,\nu} \\ \tilde{S}_{i,j}^{\mu,\nu} &= e^{-\frac{\alpha_i^\mu \alpha_j^\nu}{\alpha_i^\mu + \alpha_j^\nu} r_{AB}^2} \sum_{k,l} N_i^k N_j^l \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} \\ \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} &= \sum_{m=0}^{a_k} \sum_{n=0}^{a_l} \binom{a_k}{m} \binom{a_l}{n} x_{PA}^{a_k-m} x_{PB}^{a_l-n} \frac{\sqrt{2\pi}(m+n-1)!!}{[2(\alpha_i^\mu + \alpha_j^\nu)]^{(m+n+1)/2}} [(m+n) \text{ even}] \end{aligned}$$

Here we have used an Iverson bracket to simplify notation; this is a logical statement within square brackets which takes the value 1 if the statement is true and 0 otherwise. In this case, $[(m+n) \text{ even}] = 1$ if and only if $(m+n)$ is an even integer; otherwise it equals 0. In addition, $n!!$ indicates a double factorial: when n is even (odd), it is the product of all even (odd) numbers between 1 and n . Note that $0!! = 1$ and $(-1)!! = 1$.

A.2 Kinetic Energy Integrals

$$\begin{aligned}
t_{\mu\nu} &= \sum_{i=1}^{M_\mu} \sum_{j=1}^{M_\nu} \lambda_i^\mu \lambda_j^\nu \tilde{t}_{i,j}^{\mu,\nu} \\
\tilde{t}_{i,j}^{\mu,\nu} &= e^{-\frac{\alpha_i^\mu \alpha_j^\nu}{\alpha_i^\mu + \alpha_j^\nu} r_{AB}^2} \sum_{k,l} N_i^k N_j^l \left[\alpha_j^\nu (2(a_l + b_l + c_l) + 3) \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} \right. \\
&\quad - 2(\alpha_j^\nu)^2 \left(\tilde{X}_{i,j,a_k,a_l+2}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} + \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l+2}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} \right. \\
&\quad \left. \left. + \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l+2}^{\mu,\nu} \right) - \frac{1}{2} a_l (a_l - 1) \tilde{X}_{i,j,a_k,a_l-2}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} \right. \\
&\quad \left. - \frac{1}{2} b_l (b_l - 1) \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l-2}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l}^{\mu,\nu} - \frac{1}{2} c_l (c_l - 1) \tilde{X}_{i,j,a_k,a_l}^{\mu,\nu} \tilde{Y}_{i,j,b_k,b_l}^{\mu,\nu} \tilde{Z}_{i,j,c_k,c_l-2}^{\mu,\nu} \right]
\end{aligned}$$

A.3 Nuclear Attraction Integrals

$$\begin{aligned}
v_{\mu\nu}^D &= \sum_{i=1}^{M_\mu} \sum_{j=1}^{M_\nu} \lambda_i^\mu \lambda_j^\nu \tilde{v}_{i,j}^{\mu,\nu,D} \\
\tilde{v}_{i,j}^{\mu,\nu,D} &= \frac{-2\pi Z_D}{\alpha_i + \alpha_j} e^{-\frac{\alpha_i^\mu \alpha_j^\nu}{\alpha_i^\mu + \alpha_j^\nu} r_{AB}^2} \sum_{k,l} N_i^k N_j^l \sum_{m_x=0}^{a_k} \sum_{n_x=0}^{a_l} \sum_{r_x=0}^{\lfloor \frac{1}{2}(m_x+n_x) \rfloor} \sum_{s_x=0}^{\lfloor \frac{1}{2}(m_x+n_x-2r_x) \rfloor} A_{m_x,n_x,r_x,s_x}^{x,a_k,a_l} \\
&\quad \times \sum_{m_y=0}^{b_k} \sum_{n_y=0}^{b_l} \sum_{r_y=0}^{\lfloor \frac{1}{2}(m_y+n_y) \rfloor} \sum_{s_y=0}^{\lfloor \frac{1}{2}(m_y+n_y-2r_y) \rfloor} A_{m_y,n_y,r_y,s_y}^{y,b_k,b_l} \\
&\quad \times \sum_{m_z=0}^{c_k} \sum_{n_z=0}^{c_l} \sum_{r_z=0}^{\lfloor \frac{1}{2}(m_z+n_z) \rfloor} \sum_{s_z=0}^{\lfloor \frac{1}{2}(m_z+n_z-2r_z) \rfloor} A_{m_z,n_z,r_z,s_z}^{z,c_k,c_l} \\
&\quad \times F_v((\alpha_i + \alpha_j) r_{PD}^2) \\
v &= (m_x + n_x - 2r_x - s_x) + (m_y + n_y - 2r_y - s_y) + (m_z + n_z - 2r_z - s_z) \\
A_{m_x,n_x,r_x,s_x}^{x,a_k,a_l} &= \binom{a_k}{m_x} \binom{a_l}{n_x} (x_{PA})^{a_k-m_x} (x_{PB})^{a_l-n_x} \frac{(-1)^{m_x+n_x+s_x} (x_{PD})^{m_x+n_x-2r_x-2s_x}}{(4(\alpha_i + \alpha_j))^{r_x+s_x}} \\
&\quad \times \frac{(m_x + n_x)!}{r_x! s_x! (m_x + n_x - 2r_x - 2s_x)!}
\end{aligned}$$

$F_v((\alpha_i + \alpha_j) r_{PD}^2)$ is the Boys function, a special case of the Kummer confluent hypergeometric function. It is defined as

$$F_v(u) = \int_0^1 dt t^{2v} e^{-ut^2} \quad (77)$$

and standard implementations for evaluating the function can be found, for example in the `scipy.special` python package.

A.4 Electron Repulsion Integrals

$$\begin{aligned}
G_{\mu\nu\mu'\nu'} &= \sum_{i=1}^{M_\mu} \sum_{j=1}^{M_\nu} \sum_{i'=1}^{M_{\mu'}} \sum_{j'=1}^{M_{\nu'}} \lambda_i^\mu \lambda_j^\nu \lambda_{i'}^{\mu'} \lambda_{j'}^{\nu'} \tilde{G}_{i,j,i',j'}^{\mu\nu\mu'\nu'} \\
\tilde{G}_{i,j,i',j'}^{\mu\nu\mu'\nu'} &= \frac{2 e^{-\frac{\alpha_i \alpha_j}{\alpha_i + \alpha_j} r_{AB}^2} e^{-\frac{\alpha_{i'} \alpha_{j'}}{\alpha_{i'} + \alpha_{j'}} r_{A'B'}^2}}{(\alpha_i + \alpha_j)(\alpha_{i'} + \alpha_{j'})} \left(\frac{\pi^5}{\alpha_i + \alpha_j + \alpha_{i'} + \alpha_{j'}} \right)^{1/2} \sum_{k,l,k',l'} N_i^k N_j^l N_{i'}^{k'} N_{j'}^{l'} \\
&\times \sum_{m_x=0}^{a_k} \sum_{n_x=0}^{a_l} \sum_{r_x=0}^{\lfloor \frac{1}{2}(m_x+n_x) \rfloor} \sum_{m'_x=0}^{a_{k'}} \sum_{n'_x=0}^{a_{l'}} \sum_{r'_x=0}^{\lfloor \frac{1}{2}(m'_x+n'_x) \rfloor} \sum_{s_x=0}^{\lfloor \frac{1}{2}(m_x+n_x+m'_x+n'_x-2r_x-2r'_x) \rfloor} g_{m_x, n_x, r_x, m'_x, n'_x, r'_x, s_x}^{x, a_k, a_l, a_{k'}, a_{l'}} \\
&\times \sum_{m_y=0}^{b_k} \sum_{n_y=0}^{b_l} \sum_{r_y=0}^{\lfloor \frac{1}{2}(m_y+n_y) \rfloor} \sum_{m'_y=0}^{b_{k'}} \sum_{n'_y=0}^{b_{l'}} \sum_{r'_y=0}^{\lfloor \frac{1}{2}(m'_y+n'_y) \rfloor} \sum_{s_y=0}^{\lfloor \frac{1}{2}(m_y+n_y+m'_y+n'_y-2r_y-2r'_y) \rfloor} g_{m_y, n_y, r_y, m'_y, n'_y, r'_y, s_y}^{y, b_k, b_l, b_{k'}, b_{l'}} \\
&\times \sum_{m_z=0}^{c_k} \sum_{n_z=0}^{c_l} \sum_{r_z=0}^{\lfloor \frac{1}{2}(m_z+n_z) \rfloor} \sum_{m'_z=0}^{c_{k'}} \sum_{n'_z=0}^{c_{l'}} \sum_{r'_z=0}^{\lfloor \frac{1}{2}(m'_z+n'_z) \rfloor} \sum_{s_z=0}^{\lfloor \frac{1}{2}(m_z+n_z+m'_z+n'_z-2r_z-2r'_z) \rfloor} g_{m_z, n_z, r_z, m'_z, n'_z, r'_z, s_z}^{z, c_k, c_l, c_{k'}, c_{l'}} \\
&\times F_v \left(\frac{(\alpha_i + \alpha_j)(\alpha_{i'} + \alpha_{j'})}{\alpha_i + \alpha_j + \alpha_{i'} + \alpha_{j'}} r_{PP'}^2 \right) \\
v &= (m_x + n_x + m'_x + n'_x - 2(r_x + r'_x) - s_x) + (m_y + n_y + m'_y + n'_y - 2(r_y + r'_y) - s_y) \\
&+ (m_z + n_z + m'_z + n'_z - 2(r_z + r'_z) - s_z) \\
g_{m_x, n_x, r_x, m'_x, n'_x, r'_x, s_x}^{x, a_k, a_l, a_{k'}, a_{l'}} &= \binom{a_k}{m_x} \binom{a_l}{n_x} \binom{a_{k'}}{m'_x} \binom{a_{l'}}{n'_x} (\alpha_i + \alpha_j)^{r_x - (m_x + n_x)} (\alpha_{i'} + \alpha_{j'})^{r'_x - (m'_x + n'_x)} \\
&\times (x_{PA})^{a_k - m_x} (x_{PB})^{a_l - n_x} (x_{P'A'})^{a_{k'} - m'_x} (x_{P'B'})^{a_{l'} - n'_x} (x_{PP'})^{m_x + n_x + m'_x + n'_x - 2(r_x + r'_x + s_x)} \\
&\times \frac{(m_x + n_x)! (m'_x + n'_x)! (m_x + n_x + m'_x + n'_x - 2(r_x + r'_x))!}{s_x! r_x! r'_x! (m_x + n_x - 2r_x)! (m'_x + n'_x - 2r'_x)! (m_x + n_x + m'_x + n'_x - 2(r_x + r'_x + s_x))!} \\
&\times \frac{(-1)^{m_x + n_x + s_x}}{4^{r_x + r'_x + s_x}} \left(\frac{[(\alpha_i + \alpha_j)(\alpha_{i'} + \alpha_{j'})]}{\alpha_i + \alpha_j + \alpha_{i'} + \alpha_{j'}} \right)^{m_x + n_x + m'_x + n'_x - 2(r_x + r'_x) - s_x}
\end{aligned}$$

In all of the above equations, unprimed indices are associated with electron 1 and primed indices are associated with electron 2, and $F_v(u)$ is again the Boys function, defined above.

As one can see, the number of two-electron integrals can become quite large if many basis functions are used. However, this number can be reduced by exploiting the symmetry of these integrals. For real orbitals labeled with i, j, k, l , one can show the following identity:

$$\langle ij|kl \rangle = \langle ji|lk \rangle = \langle kl|ij \rangle = \langle lk|ji \rangle = \langle kj|il \rangle = \langle li|jk \rangle = \langle il|kj \rangle = \langle jk|li \rangle \quad (78)$$

We only have to save one of these eight integrals and thus can reduce the storage requirements by a factor of 8. To this end, we use a unique ordering of the indices in the two-electron integrals which enforces this symmetry:

- The second index j must be smaller than the first index i .
- The fourth index l must be smaller than the third index k .
- The inequality $\frac{k(k+1)}{2} + l < \frac{i(i+1)}{2} + j$ must be satisfied.