

# Documentación básica del método de Hartree–Fock restringido (RHF)

## Resumen

Este documento describe de manera concisa la implementación del método de Hartree–Fock restringido (RHF) en el código desarrollado para moléculas diatómicas con una base mínima tipo `s`. Se presentan los componentes principales del algoritmo, la estructura de módulos para el cálculo de integrales, y el procedimiento de autoconsistencia (SCF). El objetivo es proporcionar una referencia breve que documente las decisiones de diseño sin entrar en los desarrollos matemáticos de las integrales gaussianas.

## 1. Introducción

El método de Hartree–Fock (HF) busca obtener una aproximación determinantal a la función de onda de un sistema electrónico, minimizando la energía total bajo la restricción de ortonormalidad de los orbitales. La versión restringida (RHF) asume que los electrones de espín  $\alpha$  y  $\beta$  ocupan el mismo conjunto de orbitales espaciales, simplificando el cálculo.

En esta implementación, el procedimiento se desarrolla para una base mínima de funciones tipo Gaussiana, suficiente para moléculas diatómicas como  $\text{H}_2$  o  $\text{HeH}^+$ .

## 2. Estructura del código

El cálculo principal se realiza en el módulo `rhf_s_old.py`, apoyado por el módulo `s_integrals.py`, que contiene las rutinas para el cálculo de los integrales de una base `s`-tipo.

### 2.1. Cálculo de integrales atómicas

El módulo `s_integrals.py` construye los arreglos de integrales requeridos para el procedimiento SCF:

- **Matriz de traslape**  $S_{\mu\nu} = \langle \phi_\mu | \phi_\nu \rangle$
- **Matriz cinética**  $T_{\mu\nu} = \langle \phi_\mu | -\frac{1}{2} \nabla^2 | \phi_\nu \rangle$
- **Atracción nuclear**  $V_{\mu\nu} = \langle \phi_\mu | \sum_A \frac{-Z_A}{r_A} | \phi_\nu \rangle$
- **Integrales bielectrónicas**  $(\mu\nu|\lambda\sigma)$

Cada integral se calcula como suma de contribuciones entre primitivas gaussianas de tipo  $(\alpha, d)$ , y las integrales bielectrónicas se almacenan en forma *sparse* mediante un diccionario indexado por claves canónicas que respetan las simetrías de permutación:

$$(\mu\nu|\lambda\sigma) = (\nu\mu|\lambda\sigma) = (\mu\nu|\sigma\lambda) = \dots$$

## 2.2. Construcción del Hamiltoniano

Se define el Hamiltoniano de núcleo fijo:

$$H_{\mu\nu}^{\text{core}} = T_{\mu\nu} + V_{\mu\nu}.$$

Este operador se usa para obtener la energía de referencia (guess inicial) y la energía electrónica total:

$$E_{\text{elec}} = \sum_{\mu\nu} P_{\mu\nu} (H_{\mu\nu}^{\text{core}} + \frac{1}{2} F_{\mu\nu}),$$

donde  $P_{\mu\nu}$  es la matriz de densidad.

## 3. Algoritmo SCF (RHF)

El procedimiento de autoconsistencia (SCF) implementado sigue los pasos clásicos:

1. **Inicialización de matrices:** se calculan  $S, T, V, H_{\text{core}}$  y el diccionario de integrales bi-electrónicas `eri_dict`.
2. **Ortonormalización simétrica:** se obtiene  $X = S^{-1/2}$  mediante `fractional_matrix_power(S, -0.5)`.
3. **Conjetura inicial:** matriz de densidad  $P = 0$ .
4. **Bucle SCF:**

a) Construcción del operador de Fock:

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} + \sum_{\lambda\sigma} P_{\lambda\sigma} [(\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma)].$$

b) Transformación a la base ortonormal:  $F' = X^T F X$ .

c) Diagonalización de  $F'$  para obtener los orbitales moleculares  $C'$  y energías  $\epsilon$ .

d) Retrotransformación:  $C = X C'$ .

e) Construcción de la nueva densidad:

$$P_{\mu\nu} = 2 \sum_i^{\text{occ}} C_{\mu i} C_{\nu i}.$$

f) Cálculo de la energía total y prueba de convergencia ( $\Delta P_{\text{rms}} < 10^{-6}$ ).

El ciclo termina cuando la matriz de densidad converge. La energía total final se obtiene como:

$$E_{\text{total}} = E_{\text{elec}} + E_{\text{nuc}}.$$

## 4. Módulos auxiliares

El código incluye utilidades adicionales:

- `compute_nuclear_repulsion_energy`: calcula la energía de repulsión nuclear generalizada.
- `primitives_to_ao_list`, `ao_centers_from_pos`, `atom_of_mu_from_primitives`: funciones de mapeo entre primitivas, átomos y orbitales.
- `print_final_results`: genera un resumen final de la energía y los orbitales convergidos.

## 5. Notas sobre implementación

- Las integrales están programadas para una base **s**-tipo; la extensión a **p** u orbitales de mayor momento angular requiere nuevos tensores de integrales.
- La estructura `eri_dict` reduce significativamente el costo en memoria respecto a un tensor  $N^4$ .
- El algoritmo usa **ortonormalización simétrica**, que es estable incluso cuando  $S$  es mal condicionado.

## Conclusión

Este módulo constituye una implementación autocontenida y didáctica del método de Hartree–Fock restringido. Aun en su forma básica, permite explorar los principios del método SCF, el papel de las integrales atómicas, y la construcción del operador de Fock. Futuras extensiones pueden incluir: uso de bases extendidas (STO-3G, 6-31G), incorporación de integrales  $p$ -tipo, y el desarrollo de módulos post-Hartree–Fock.