

- 电子哈密顿量
  - 总哈密顿量
  - BO近似，绝热近似
  - 单电子哈密顿量
- 波函数
- 能量表达式
- 能量最小化
- 编程实现
  - 正则HF方程
  - Roothaan-Fall 方程
  - 自洽场求解 (SCF)
  - 推广到具体的自旋轨道
- References
  - 理论推导
  - 代码实践

# 电子哈密顿量

## 总哈密顿量

分子体系的总哈密顿量表示为

$$\hat{H}_{tot} = \hat{T}_n + \hat{T}_e + \hat{V}_{nn} + \hat{V}_{ne} + \hat{V}_{ee}$$

其中包含两类五项：

- 动能：核，电子
- 势能：核-核，电子-电子，核-电子

## BO近似，绝热近似

- 电子速度 >> 核运动速度，核相对静止（核动能为0）
- 核感受的电子势能平均化

## 单电子哈密顿量

$$\begin{aligned}\hat{H}_{el} &= \hat{H}_{tot} - \hat{T}_n = \hat{T}_e + \hat{V}_{nn} + \hat{V}_{ne} + \hat{V}_{ee} \\ &= \sum_i -\frac{1}{2} \nabla_i^2 - \sum_i \sum_A \frac{Z_A}{|r_i - R_A|} + \frac{1}{2} \sum_i \sum_j \frac{1}{|r_i - r_j|} + \hat{V}_{nn}\end{aligned}$$

其中  $-\frac{1}{2} \nabla_i^2$  为动能算符， $Z_A$  为第  $A$  个原子核的电荷量（正数）， $R_A$  为原子核的空间坐标， $r_i$  为第  $i$  个电子的空间坐标，第三项为任意两个电子间的排斥能，最后一项为核间互斥能，体系给定它就是一个常数，后续推导可以暂时忽略此项。

将多体问题分解为求解单电子本征值方程：

$$\hat{H}_{el} |\Psi_i^{el}\rangle = E_{el} |\Psi_i^{el}\rangle$$

Hartree-Fock 目的：选择一个最好的组态，使得体系能量最小化。关键在于如何处理平均场即  $\hat{V}_{ee}$ 。

## 波函数

Slater行列式表示

- 交换反对称
- Pauli不相容原理

双电子实例：

$$\Psi(\phi_1, \phi_2) = \begin{vmatrix} \phi_1(x_1) & \phi_1(x_2) \\ \phi_2(x_1) & \phi_2(x_2) \end{vmatrix} = \frac{1}{\sqrt{2}} [\phi_1(x_1)\phi_2(x_2) - \phi_1(x_2)\phi_2(x_1)]$$

其中  $\phi_i(x_j)$  表示第  $j$  个电子在第  $i$  个分子轨道状态,  $x_i = (r_i, w_i)$

## 能量表达式

以2电子体系举例

能量表达式：展开12项，最后剩余4项

一些简化的记号

$$\begin{aligned} E &= \langle \Psi | \hat{H} | \Psi \rangle \\ &= \sum_i \langle i | h | i \rangle + \frac{1}{2} \sum_i \sum_j \langle ij | ij \rangle \end{aligned}$$

其中单电子积分记号为：

$$\langle i | h | j \rangle = \int \phi_i^*(x) h(r) \phi_j(x) dx$$

双电子积分记号为：

$$\langle ij|kl\rangle = \iint \phi_i^*(x_1)\phi_j^*(x_2)\frac{1}{|r_1-r_2|}\phi_k(x_1)\phi_l(x_2)dx_1dx_2$$
$$\langle ij||ij\rangle = \langle ij|ij\rangle - \langle ij||ji\rangle$$

## 能量最小化

变分法，用拉格朗日乘子法保证约束条件，分子轨道满足正交归一

限制条件： $\langle i|j\rangle = \delta_{ij} \Rightarrow \langle i|j\rangle - \delta_{ij} = 0$

拉格朗日乘子法：

$$L = E - \sum_{ij} \lambda_{ij} (\langle i|j\rangle - \delta_{ij})$$

扰动来获取对应的结果

改变第  $m$  个轨道， $\phi_m = (1 + \delta)\phi_m$ ，简洁记为  $m = m + \delta m$ 。

极值条件： $\frac{\partial L}{\partial m} = 0$  等价于

$$\delta L = L(m = m + \delta m) - L(m) = 0$$

变分：针对单电子，双电子，拉格朗日系数分别做微分，最后合并结果（仅考虑一阶贡献）。最后得到HF方程：

$$\hat{f}_i \phi_m = \sum_i \lambda_{mi} \phi_i$$

其中 Fock 算符为  $\hat{f}_i = \hat{h}_i + \hat{J} - \hat{K}$ ，式子中  $\hat{J}$  为库伦算符， $\hat{K}$  为交换算符（强定义，仅仅为了一致性），其具体定义如下：

$$\hat{J}g_m(x_2) = \sum_i \int \phi_i^*(x_1) \frac{1}{r_{12}} \phi_i(x_1) dx_1 g_m(x_2)$$

$$\hat{K}g_m(x_2) = \sum_i \int \phi_i^*(x_1) \frac{1}{r_{12}} \phi_i(x_2) dx_1 g_m(x_1)$$

## 编程实现

先进行HF方程的正则化，再变换为容易被计算机求解的矩阵形式。主要是为了展示推导的思路，具体实现时考虑了轨道自旋情况，最终Fock算符和能量表达式的系数会有所变化。

## 正则HF方程

通过酉变换和对角化将其变为正则HF方程

选一个能够对角化  $\lambda$  系数矩阵的酉矩阵  $U$ ，即  $U\lambda U^{-1}$  为对角矩阵，从而将波函数转换为  $U\Phi$ ，其具体推导如下：

$$F\Phi = \lambda\Phi$$

$$UF\Phi = U\lambda U^{-1}U\Phi$$

$$F(U\Phi) = (U\lambda U^{-1})(U\Phi)$$

$$F\Phi' = \epsilon\Phi'$$

因此能够得到正则HF方程：

$$\hat{f}_i\phi_i = \epsilon_i\phi_i$$

标准的本征值问题。

# Roothaan-Fall 方程

若想在计算机上实现，必须将分子轨道用原子轨道展开，即分子轨道是原子轨道的线性组合： $\phi_i = \sum_u C_{ui} \chi_u$ 。将该式子代入正则HF方程，并同时左乘一个基底函数并积分，即可得到Roothaan-Fall 方程：

$$FC = SCE$$

其中

- F是 Fock 矩阵： $F_{uv} = \langle \chi_u | \hat{f} | \chi_v \rangle = h_{uv} + \sum_{\alpha\beta} D_{\alpha\beta} [\langle \alpha u | \beta v \rangle - \langle \alpha u | v \beta \rangle]$ ，其中密度矩阵  $D_{uv} = \sum_i C_{ui}^* C_{vi}$
- S是重叠矩阵： $S_{uv} = \langle \chi_u | \chi_v \rangle$ ，原子轨道不保证正交归一
- C是系数矩阵
- E是对角矩阵，代表每个分子轨道的能量

## 自洽场求解（SCF）

存在循环依赖：F->D->C->F

迭代求解算法

- 给定原子基组  $\{\chi_i\}$ ，计算重叠矩阵  $S$ ， $h_{uv}$ ，双电子积分
- 猜一个  $C^{(0)}$ ，进而计算出  $D^{(0)}$ ， $F^{(0)}$
- 对角化  $FC = SCE$ ，得到  $C^{(1)}$
- 重复第2,3步骤，直至收敛（D前后两次变量低于某个阈值）

对角化的方程并非为标准的特征值方程，因此对方程  $FC = SCE$  进行如下变换，

$$\begin{aligned}
FC &= SCE \\
FS^{-\frac{1}{2}}S^{\frac{1}{2}}C &= S^{\frac{1}{2}}S^{\frac{1}{2}}CE \\
(S^{-\frac{1}{2}}FS^{-\frac{1}{2}})(S^{\frac{1}{2}}C) &= (S^{\frac{1}{2}}C)E \\
F'C' &= C'E
\end{aligned}$$

可以采用如下步骤实现  $FC = SCE$ :

1. 计算  $S^{-\frac{1}{2}}$
2. 计算  $F' = S^{-\frac{1}{2}}FS^{-\frac{1}{2}}$
3. 对角化  $F'C' = C'E$ , 求得  $C'$
4. 计算  $C = S^{-\frac{1}{2}}C'$

## 推广到具体的自旋轨道

对应的编程实现是推广到了自旋轨道，其中Fock算符，能量最终表达式需要重新推导，推导时需要加上自旋，然后先通过积分消去自旋，其后思路和前文提到的一般形式推导一样。具体

考虑自旋时的方程推导，总体思路是先消去自旋部分，再积分消掉空间部分，进行后续推导。

自旋轨道拆分为两个部分，空间坐标  $\varphi$  和自旋部分  $\alpha, \beta$ 。

$$\phi_i(x) = \begin{cases} \varphi_j(r)\alpha(w) \\ \varphi_j(r)\beta(w) \end{cases}$$

我们从正则HF方程开始做自旋轨道的推广，先假设时自旋为  $\alpha$  ( $\beta$  结果也一样)：

$$f(x_1)\phi_i(x_1) = \epsilon_i\phi_i(x_1)$$

$$f(x_1)\varphi_j(r_1)\alpha(w_1) = \epsilon_i\varphi_i(r_1)\alpha(w_1)$$

通过积分消除自旋，即左乘  $\alpha(w_1)$  并两边积分，同时利用自旋的正交归一性化简等式右侧：

$$\begin{aligned}\int \alpha^*(w_1)f(x_1)\varphi_j(r_1)\alpha(w_1)dw_1 &= \int \alpha_i^*(w_1)\epsilon_i\varphi_i(r_1)\alpha(w_1)dw_1 \\ \int \alpha^*(w_1)f(x_1)\varphi_j(r_1)\alpha(w_1)dw_1 &= \epsilon_i\varphi_i(r_1)\end{aligned}$$

将Fock算符具体形式展开，得到下式：

$$\begin{aligned}\left[\int \alpha^*(w_1)f(x_1)\alpha(w_1)dw_1\right]\varphi_j(r_1) &= \left[\int \alpha^*(w_1)h(r_1)\alpha(w_1)dw_1\right]\varphi_j(r_1) \\ &+ \left[\sum_c \int \alpha^*(w_1)\phi_c^*(x_2)r_{12}^{-1}\phi_c(x_2)\alpha(w_1)dw_1dx_2\right]\varphi_j(r_1) \\ &- \left[\sum_c \int \alpha^*(w_1)\phi_c^*(x_2)r_{12}^{-1}\phi_c(x_1)\alpha(w_2)dw_1dx_2\right]\varphi_j(r_2)\end{aligned}$$

由于闭壳层中自旋向上和向下的轨道数目相等，因此可以将上式中所有的自旋轨道都拆分为2部分，并利用正交归一性进行化简：



$$\begin{aligned}
& \left[ \int \alpha^*(w_1) f(x_1) \alpha(w_1) dw_1 \right] \varphi_j(r_1) = h(r_1) \varphi_j(r_1) \\
& + \left[ \sum_c^{N/2} \int \alpha^*(w_1) \varphi_c^*(r_2) \alpha^*(w_2) r_{12}^{-1} \varphi_c(r_2) \alpha(w_2) \alpha(w_1) dw_1 dw_2 dr_2 \right] \varphi_j(r_1) \\
& + \left[ \sum_c^{N/2} \int \alpha^*(w_1) \varphi_c^*(r_2) \beta^*(w_2) r_{12}^{-1} \varphi_c(r_2) \beta(w_2) \alpha(w_1) dw_1 dw_2 dr_2 \right] \varphi_j(r_1) \\
& - \left[ \sum_c^{N/2} \int \alpha^*(w_1) \varphi_c^*(r_2) \alpha^*(w_2) r_{12}^{-1} \varphi_c(r_1) \alpha(w_1) \alpha(w_2) dw_1 dw_2 dr_2 \right] \varphi_j(r_2) \\
& - \left[ \sum_c^{N/2} \int \alpha^*(w_1) \varphi_c^*(r_2) \beta^*(w_2) r_{12}^{-1} \varphi_c(r_1) \beta(w_1) \alpha(w_2) dw_1 dw_2 dr_2 \right] \varphi_j(r_2) \\
& = h(r_1) \varphi_j(r_1) + 2 \left[ \sum_c^{N/2} \int \varphi_c^*(r_2) r_{12}^{-1} \varphi_c(r_2) dr_2 \right] \varphi_j(r_1) \\
& - \left[ \sum_c^{N/2} \int \varphi_c^*(r_2) r_{12}^{-1} \varphi_c(r_1) dr_2 \right] \varphi_j(r_2)
\end{aligned}$$

令  $f(r_1) \equiv \int \alpha^*(w_1) f(x_1) \alpha(w_1) dw_1$  (对  $w_1$  全空间积分, 所以只和  $r_1$  相关), 并且  $\epsilon_i$  为自旋轨道的能量, 与对应空间轨道能量相同, 因此上式等价于:

$$f(r_1) \varphi_j(r_1) = \epsilon_j \varphi_j(r_1)$$

同时可以发现, 可以定义和前面通用的库伦  $\hat{J}$  和交换算符  $\hat{K}$ , 他们形式基本一致, 只不过此处将自旋轨道换成空间轨道。因此闭壳层的 Fock 算符形式为:

$$f(r_1) = h(r_1) + 2\hat{J}_a - \hat{K}_a$$

$$\hat{J}_a \varphi_j(r_1) = \sum_c^{N/2} \int \varphi_c^*(r_2) r_{12}^{-1} \varphi_c(r_2) dr_2 \varphi_j(r_1)$$

$$\hat{K}_a \varphi_j(r_1) = \sum_c^{N/2} \int \varphi_c^*(r_2) r_{12}^{-1} \varphi_j(r_2) dr_2 \varphi_c(r_1)$$

已知Fock算符形式，很容易推导出  $F_{uv} = \langle \chi_u | \hat{f} | \chi_v \rangle = h_{uv} + \sum_{\alpha\beta} D_{\alpha\beta} [2\langle \alpha u | \beta v \rangle - \langle \alpha u | v \beta \rangle]$ ，其中密度矩阵  $D_{uv} = \sum_i C_{ui}^* C_{vi}$ ，此处也是库伦势能前多了系数2，本质是由Fock算符影响的。

同理，用自旋轨道作为波函数去推导体系能量，先将其转为分子轨道（积分消去自旋部分，对自旋分类讨论，用正交归一性化简），再用原子基组展开，就能得到RHF下的能量具体表达式。

## References

对于量子化学初学者（原计算机选手，数学基础仅有高数，线代，概率论，离散数学）想要快速入门理解，我采用的路线是忽略大量的基础数学知识（一开始就看这个很容易觉得无聊从而放弃，别问我咋知道的），直接看hartree-fock的推导视频，在看视频中将不懂得基础概念和数学方法粗略了解下（先掌握脉络，后面再深入），然后跟着推公式，自己整理笔记，发现不通畅时回去看视频或者看书/文章，解决一个个的问题。最后是将公式转换为代码，来检验自己的推导是否正确，是否真的理解学习内容。

这么一套下来，在学中练，练中学，理论与实践并进，同时目标明确，反馈及时，能够促进思考与实践效率。

## 理论推导

[量子化学入门【已完结】](#)：白板推导教学，强烈推荐新手小白观看学习

[Hartree-Fock超详细推导](#)：文字版，非常详细，可对照视频看，但不包括自旋部分

[从Hartree-Fock方程推导Roothaan方程](#)：白板推导，包括了自旋部分，能得到严格的RHF的求解公式

《Modern quantum chemistry introduction to advanced electronic structure theory》 Attila Szabo，适合入门观看，HF部分教你实现Hartree-Fock代码，大多数的网络资料均来源于此，包括本文

《量子化学中册》徐光宪，非常详实，想了解数学和技术细节的冲他，但初学最好和Szabo，上面推荐的视频一块食用

《Quantum Chemistry》：Levine，比较适合入门选手

## 代码实践

[hf-tutorial](#)：提供一个测试框架，让你填空实现RHF，各类积分，重叠矩阵，哈密顿矩阵都作为输入，你无需关心，只要实现最核心的SCF算法，同时提供了一份已有的解决方案作为参考，非常适合检验自己的学习成果。

[Psi4Numpy](#)：基于Psi4和Numpy实现各类量化程序的教程

[xDH在python下实现的简易教程](#)：量化介绍及其实现原理实践