

- Roughly speaking, in Davidson algorithm, one needs to implement the function which maps an input FCI vector $|\Psi\rangle$ into $\hat{H}|\Psi\rangle$.
- We need to first think about how to represent & store the FCI vectors $|\Psi\rangle$ numerically. One can think of a basis consisting of determinants $|\Phi\rangle$, and we can store the coefficients $\langle\Phi|\Psi\rangle$. Let's take H_6 as an example. You have suggested to enumerate all the α and β determinants (C_6^3 each), which is reasonable. We need to further notice that suitable $|\Phi\rangle$ should be a product of the α and β determinants, $|\Phi_{IJ}\rangle = |\Phi_I^\alpha\rangle \otimes |\Phi_J^\beta\rangle$. So if there are C_6^3 each, the total number of possible FCI vectors spanning the variational space should be $(C_6^3)^2 = 400$. Numerically this can be a `400 × 1 np.array`.
- We then think of how to perform matrix-vector multiplications. Previously we just build the 400×400 Hamiltonian matrix $\langle\Phi|\hat{H}|\Phi'\rangle$ and store it. However, when system size becomes large, this is no longer feasible. An observation is that Hamiltonian matrix in FCI space is in fact very sparse – we are told by Slater-Condon rules that if the determinants differ by more than 2 excitations, then the matrix element between them is zero, so storing everything is not efficient.
- An approach is to get rid of the FCI matrix representation of the Hamiltonian and do everything directly under second-quantization. We expand

$$\hat{H}|\Psi\rangle = \sum_I \langle\Phi_I|\Psi\rangle \hat{H}|\Phi_I\rangle = \sum_I C_I \hat{H}|\Phi_I\rangle \quad (1)$$

We can then think of applying \hat{H} to each $|\Phi_I\rangle$ determinant. For example for 1-body $\hat{H} = \sum_{\mu\nu} h_{\mu\nu} c_\mu^\dagger c_\nu$, we have

$$\hat{H}|\Phi_I\rangle = \sum_{\mu\nu} h_{\mu\nu} c_\mu^\dagger c_\nu |\Phi_I\rangle = \sum_{\mu\nu} (-1)^{\sigma(I,\mu,\nu)} h_{\mu\nu} |\Phi_{f(I,\mu,\nu)}\rangle \quad (2)$$

A key observation is that when we apply $c_\mu^\dagger c_\nu$ to a determinant $|\Phi_I\rangle$, it gives us some other determinant $|\Phi_{f(I,\mu,\nu)}\rangle$ with a phase factor $(-1)^{\sigma(I,\mu,\nu)}$. Here f is a function that takes numbers I, μ, ν and outputs a number $J = f(I, \mu, \nu)$ to indicate that we have reached the J -th determinant. An intuitive thinking is that for H_6 , we can start from the zeroth ($I = 0$) determinant $|000111\rangle_\alpha \otimes |000111\rangle_\beta$, remove the electron in the zeroth α orbital ($\nu = 0$) and add it to the third orbital ($\mu = 3$), this actually gives us this determinant $|001110\rangle \otimes |000111\rangle$,

which is the first ($J = 1$). So $f(0, 3, 0) = 1$. You may notice that when we talk about the “first”, “second” or the “zeroth” determinants, we are actually referring to the “addresses” of the determinants, i.e. address is something one use to “label” the determinants.

- If we know the functions $f(I, \mu, \nu)$ and $\sigma(I, \mu, \nu)$, it is then simple to compute

$$\hat{H}|\Psi\rangle = \sum_I C_I \hat{H}|\Phi_I\rangle = \sum_I C_I \sum_{\mu\nu} (-1)^{\sigma(I, \mu, \nu)} h_{\mu\nu} |\Phi_{f(I, \mu, \nu)}\rangle \quad (3)$$

as a summation. Note that here only the molecular integrals appear (not the FCI Hamiltonian matrices!). We need to be able to compute $f(I, \mu, \nu)$ and $\sigma(I, \mu, \nu)$, and this can be done by some binary operations. I would like to recommend Chapter 11.8 of the book *Molecular Electronic-Structure Theory* for detailed explanations. Also, maybe you can look at the `fci.cistring.genlinkstr_index` function in PySCF for more hints.

A. general N -electron terms

- The above describes how to “contract” 1-electron operators to a given FCI state without explicit dense matrix representations, and our task now is to generalize to the N -electron case.
- The most “intellectually efficient” way is to think of the decomposition of N -electron excitations into individual 1-electron excitations (which is also known as the resolution of identity approach). For example we consider a 2-electron operator

$$\hat{V} = \frac{1}{2} \sum_{pqrs} \langle pq|rs\rangle c_p^\dagger c_q^\dagger c_s c_r \quad (4)$$

where p, q, r, s are **spin orbital** labels. Note that the following is an identity:

$$c_p^\dagger c_q^\dagger c_s c_r = -c_p^\dagger (\delta_{qr} - c_r c_q^\dagger) c_s = (c_p^\dagger c_r) (c_q^\dagger c_s) - c_p^\dagger c_s \delta_{qr} \quad (5)$$

which means that \hat{V} can be decomposed into

$$\hat{V} = \frac{1}{2} \sum_{pqrs} \langle pq|rs\rangle c_p^\dagger c_r c_q^\dagger c_s - \frac{1}{2} \sum_{psq} \langle pq|qs\rangle c_p^\dagger c_s \quad (6)$$

The second term is an 1-electron operator and we know how to deal with it. For the first term, since now the creation/annihilation operators are adjacent and paired, we can think

of first applying $(c_q^\dagger c_s)$ to the states $|\Phi_I\rangle$, generating $(-1)^{\sigma(I,q,s)}|\Phi_{f(I,q,s)}\rangle$, and then further applying $(c_p^\dagger c_r)$ to each $|\Phi_{f(I,q,s)}\rangle$. Explicitly,

$$\begin{aligned} \frac{1}{2} \sum_{pqrs} \langle pq|rs\rangle (c_p^\dagger c_r) (c_q^\dagger c_s) |\Phi_I\rangle &= \frac{1}{2} \sum_{pqrs} (-1)^{\sigma(I,q,s)} \langle pq|rs\rangle (c_p^\dagger c_r) |\Phi_{f(I,q,s)}\rangle \\ &= \frac{1}{2} \sum_{pqrs} (-1)^{\sigma(I,q,s)} (-1)^{\sigma(f(I,q,s),p,r)} \langle pq|rs\rangle |\Phi_{f(f(I,q,s),p,r)}\rangle \end{aligned} \quad (7)$$

We observe that only the molecular integrals $\langle pq|rs\rangle$ appear in this summation (not FCI matrices!). This scheme is “intellectually efficient” because there’s really nothing new outside of the decomposition, one already know how to compute functions σ and f in 1-electron case (basically the γ_{ij}^{JK} in your language) and can just reuse it. One may envisage that the same procedure works for N -electron operators.

- Numerically, it could be more efficient to first compress the sum

$$\frac{1}{2} \sum_{qs} \langle pq|rs\rangle (-1)^{\sigma(I,q,s)} |\Phi_{f(I,q,s)}\rangle = \sum_{\text{all possible } f(I,q,s)} D_{pr}^{f(I,q,s)} |\Phi_{f(I,q,s)}\rangle \quad (8)$$

with D as an intermediate. D can be stored as a $K \times N \times N$ array, where K is the number of possible states reachable by $f(I, q, s)$, and N is the number of orbitals. We may then add the $(c_p^\dagger c_r)$ part efficiently:

$$\begin{aligned} \frac{1}{2} \sum_{pqrs} (-1)^{\sigma(I,q,s)} \langle pq|rs\rangle (c_p^\dagger c_r) |\Phi_{f(I,q,s)}\rangle &= \sum_{pr} (c_p^\dagger c_r) \frac{1}{2} \sum_{qs} \langle pq|rs\rangle (-1)^{\sigma(I,q,s)} |\Phi_{f(I,q,s)}\rangle \\ &= \sum_{pr} \sum_K (-1)^{\sigma(K,p,r)} D_{pr}^K |\Phi_{f(K,p,r)}\rangle \end{aligned} \quad (9)$$

here we used K as an index labeling all possible $f(I, q, s)$. For a reference implementation, you can look at the `fcf.fci_slow.contract_2e` function in PySCF for more hints. The variable `t1` is similar to the D matrix here (and the E matrix in the paper you mentioned. I saved symbol E for the spin-summed excitations).

- Finally, if you are using spacial orbitals (which is likely) rather than spin orbitals, we will be decomposing N -electron spin-summed excitation operators. For example, the spin-summed 1e/2e excitations are defined as

$$\hat{E}_{pr} = \sum_{\sigma} c_{p\sigma}^\dagger c_{r\sigma}; \quad \hat{E}_{pqrs} = \frac{1}{2} \sum_{\sigma\sigma'} c_{p\sigma}^\dagger c_{q\sigma'}^\dagger c_{r\sigma'} c_{s\sigma} \quad (10)$$

In this case, please don't forget to pay attention to the extra spin indice in your derivations of the decomposition. (You may find the result to be pretty much the same, interestingly).