CHEM 8950
ADVANCE QUANTUM CHEMISTRY

This assignment is due Wednesday, April 17 at 5PM by email.

Density-Fitting

Density fitting is a way to approximate the usual two-electron integrals,

$$(pq|rs) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\frac{1}{r_{12}}\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2) \; . \tag{0.1}$$

We may also consider this electron repulsion integral as the repulsion between two generalized electron densities,

$$(pq|rs) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \rho_{pq}(\mathbf{r}_1)\frac{1}{r_{12}}\rho_{rs}(\mathbf{r}_2) \; , \tag{0.2}$$

where $\rho_{pq}(\mathbf{r}) = \phi_p(\mathbf{r})\phi_q(\mathbf{r})$ and $\rho_{rs}(\mathbf{r}) = \phi_r(\mathbf{r})\phi_s(\mathbf{r})$. The densities may be approximated using an auxiliary basis set as

$$\bar{\rho}_{pq}(\mathbf{r}) = \sum_P^{N_{fit}} d_P^{pq}\chi_P(\mathbf{r}) \; , \tag{0.3}$$

where d_P^{pq} are the fitting coefficients. This is the origin of the name "density fitting." There are various methods for obtaining these fitting coefficients. If one minimizes the following (positive definite) functional with a $\frac{1}{r_{12}}$ weight factor,

$$\Delta_{pq} = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{\left[\rho_{pq}(\mathbf{r}_1) - \bar{\rho}_{pq}(\mathbf{r}_1)\right]\left[\rho_{pq}(\mathbf{r}_2) - \bar{\rho}_{pq}(\mathbf{r}_2)\right]}{r_{12}} \tag{0.4}$$

this minimizes the error in the electric field and leads to the following fitting coefficients:

$$d_Q^{pq} = \sum_P (pq|P)[\mathbf{J}^{-1}]_{PQ} \tag{0.5}$$

where

$$(pq|P) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\frac{1}{r_{12}}\chi_P(\mathbf{r}_2) \; , \tag{0.6}$$

and

$$J_{PQ} = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \chi_P(\mathbf{r}_1)\frac{1}{r_{12}}\chi_Q(\mathbf{r}_2). \tag{0.7}$$

Thus we have replaced the usual four-index, two-electron integrals $(pq|rs)$ with

$$(pq|rs) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \rho_{pq}(\mathbf{r}_1) \frac{1}{r_{12}} \rho_{rs}(\mathbf{r}_2) \tag{0.8}$$

$$= \int d\mathbf{r}_1 \int d\mathbf{r}_2 \sum_Q d_Q^{pq} \chi_Q(\mathbf{r}_1) \frac{1}{r_{12}} \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2) \tag{0.9}$$

$$= \sum_Q d_Q^{pq} (Q|rs) \tag{0.10}$$

$$= \sum_{PQ} (pq|P) [\mathbf{J}^{-1}]_{PQ} (Q|rs) \tag{0.11}$$

To reduce the cost of constructing the four-index integrals from the three-index integrals, it can be advantageous to use a symmetric expression which splits the $\mathbf{J}^{-1}$ into a product of $\mathbf{J}^{-1/2}\mathbf{J}^{-1/2}$:

$$(pq|rs) = \sum_{PQR} (pq|P) [\mathbf{J}^{-1/2}]_{PQ} [\mathbf{J}^{-1/2}]_{QR} (R|rs). \tag{0.12}$$

This form is advantageous because it factorizes as:

$$(pq|rs) = \sum_Q \left\{ \sum_P (pq|P) [\mathbf{J}^{-1/2}]_{PQ} \right\} \left\{ \sum_R [\mathbf{J}^{-1/2}]_{QR} (R|rs) \right\} = \sum_Q b_{pq}^Q b_{rs}^Q \ , \tag{0.13}$$

where

$$b_{pq}^Q = \sum_P (pq|P) [\mathbf{J}^{-1/2}]_{PQ} \ . \tag{0.14}$$

Scaling

The most time consuming part of MP2 is the integral transformation which scales as $\mathcal{O}(N^5)$ whereas the energy computation scales $\mathcal{O}(N^4)$. The minimal integral transformation needed for MP2 is

$$(ia|jb) = \sum_{\mu\nu\rho\sigma} C_\mu^i C_\nu^a C_\rho^j C_\sigma^b (\mu\nu|\rho\sigma) \tag{0.15}$$

where i and j are occupied orbitals; a and b are virtual orbitals. If there are N_{occ} occupied orbitals and N_{vir} virtual orbitals, then there are $N_{occ}^2 N_{vir}^2$ of the required MO integrals. This transformation above would appear to require $N_{occ}^2 N_{vir}^2 N_{AO}^4$ floating-point operations. Through factorization the actual cost

is less:

$$(i\nu|\rho\sigma) = \sum_{\mu} C_{\mu}^{i}(\mu\nu|\rho\sigma) \qquad N_{occ}N_{AO}^{4} \tag{0.16}$$

$$(ia|\rho\sigma) = \sum_{\nu} C_{\nu}^{a}(i\nu|\rho\sigma) \qquad N_{occ}N_{vir}N_{AO}^{3} \tag{0.17}$$

$$(ia|j\sigma) = \sum_{\rho} C_{\rho}^{j}(ij|\rho\sigma) \qquad N_{occ}^{2}N_{vir}N_{AO}^{2} \tag{0.18}$$

$$(ia|jb) = \sum_{\sigma} C_{\sigma}^{a}(ij|j\sigma) \qquad N_{occ}^{2}N_{vir}^{2}N_{AO} \tag{0.19}$$

Since N_{AO} is the largest quantity, the first step is the most time-consuming.

With density-fitting, we factorize the equations more effectively

$$b_{i\nu}^{Q} = \sum_{\mu} C_{\mu}^{i} b_{\mu\nu}^{Q} \qquad N_{occ}N_{AO}^{2}N_{aux} \tag{0.20}$$

$$b_{ia}^{Q} = \sum_{\nu} C_{\nu}^{a} b_{i\nu}^{Q} \qquad N_{occ}N_{vir}N_{AO}N_{aux} \tag{0.21}$$

$$(ia|jb) = \sum_{Q} b_{ia}^{Q} b_{jb}^{Q} \qquad N_{occ}^{2}N_{vir}^{2}N_{aux} \tag{0.22}$$

The most expensive step of the DF transformation is the last one, with a cost of $\mathcal{O}(N_{occ}^{2}N_{vir}^{2}N_{aux})$, which is much less than $\mathcal{O}(N_{occ}N_{AO}^{4})$ from the conventional transformation.

Errors Introduced

Density-fitting does introduce an error. This output is for H_2O^+ with cc-pVTZ/cc-pVTZ-RI basis sets. Also note how much faster DF-MP2 can be when compared to conventional MP2.

```
MP2 correlation energy:     -0.2107800453
Total MP2 energy:          -75.8540977449 (HF + MP2)
MP2 total time:  12.4572 seconds.

Density fitted MP2 energy:  -0.2107758942
Total DF-MP2 energy:       -75.8540935937 (HF + DF-MP2)
DF-MP2 total time:  0.2743 seconds.

DF error:                   -0.0000041511
```

Density-Fitting MP2

Add to your existing spin-orbital MP2 code to also perform DF-MP2. Compute and print both the MP2 and DF-MP2 correlation energies as well as there difference. Include timings to see how much density-fitting improves the performance of your code.

There is one new option for this assignment:

- `df_basis` – the name of the basis set to use in the density-fitting procedure.

Some tips:

- Use something *similar* (don't blindly copy and paste) to the following line to load the DF basis. You must use '`DF_BASIS_MP2`' as that is the official Psi4 keyword.

```
df = psi4.core.BasisSet.build(self.molecule, 'DF_BASIS_MP2',
                              Settings['df_basis'],
                              puream=0)
```

- You will need an empty basis set to "trick" Psi4 to compute two- and three-center electron repulsion integrals with its four-center code.

```
zero = psi4.core.BasisSet.zero_ao_basis_set()
```

- The `MintsHelper` function `ao_eri` can accept basis set objects to use for each center. The version of this function you likely have been using assumes all the centers are the same and used the basis set provided when you constructed the `MintsHelper` object.

```
mints.ao_eri(bs1, bs2, bs3, bs4)
```

- You will solve for $\mathbf{J}^{-1/2}$ the same way you solved for $\mathbf{S}^{-1/2}$ in Hartree-Fock.
- `np.squeeze(...)` can be used to drop zero length dimensions from a NumPy array.
- The `block_tei` code you used in MP2 will have to be modified for DF-MP2.
- You may need to program the chemist-notation version of the MP2 equations.

Conventional MP2 energy equation:

$$E_{\mathrm{MP2}} = \frac{1}{4}\sum_{ijab}\frac{\langle ij||ab\rangle^2}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b} \tag{0.23}$$

$$= \frac{1}{4}\sum_{ijab}\frac{[(ia|jb)-(ib|ja)]^2}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b} \tag{0.24}$$

DF-MP2 energy equation:

$$E_{\mathrm{DF-MP2}} = \frac{1}{4}\sum_{ijab}\frac{\left[\sum_Q b_{ia}^Q b_{jb}^Q - \sum_Q b_{ib}^Q b_{ja}^Q\right]^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \tag{0.25}$$

$$= \frac{1}{4}\sum_{ij}\sum_{Q}\sum_{ab}\frac{\left[b_{ia}^Q b_{jb}^Q - b_{ib}^Q b_{ja}^Q\right]^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \tag{0.26}$$

Note: Do NOT simply code up $(ia|jb) = \sum_Q b_{ia}^Q b_{jb}^Q$ and then plug $(ia|jb)$ into your normal MP2 energy expression. Doing so will require the creation of a four-index tensor which defeats the point of density fitting. You will not receive credit for a code that does this.

```
# DO NOT DO THIS.
Iiajb = np.einsum('iaQ,jbQ->iajb', biaQ, bjbQ)
```

The point of density fitting is that you never explicitly form the two-electron integrals.

Pay close attention to the order and separation of the summations in equation 0.26 for a hint on the algorithm you should employ.

Input File

```
Settings = dict()
Settings["basis"] = "cc-pvtz"
Settings["df_basis"] = "cc-pvtz-ri"
Settings["molecule"] = """
  1 2
  O
  H 1 R
  H 1 R 2 A
  R = 0.9
  A = 104.5
  symmetry c1
"""
Settings["nalpha"] = 5
Settings["nbeta"] = 4
Settings["scf_max_iter"] = 50
```

Sample Output

```
Psi4 Hartree-Fock Energy: -75.6433176996
Psi4 MP2 Energy:          -75.8540977446


  Spin-Orbital MP2


Number of basis functions:            65
Number of molecular orbitals:         65
Number of spin orbitals:              130
    Number of occupied spin orbitals: 9
    Number of virtual spin orbitals:  121

Computing AO integrals (pq|rs) ... done in  11.7031 seconds.
Performing AO->MO integral transformation (ov|ov) ... done in  0.7402 seconds.
Anti-symmetrizing the integrals <ij||ab> = (ia|jb) - (ib|ja) ... done in  0.0023
Forming denominator (oovv) ... done in  0.0061 seconds.
Computing MP2 energy ... done in  0.0020 seconds.

MP2 correlation energy:  -0.2107800453
Total MP2 energy:       -75.8540977449 (HF + MP2)
MP2 total time:  12.4572 seconds.


  Spin-Orbital DF-MP2


Building J^(-1/2) (P|Q) ... done in  0.0373 seconds.
Building (pq|P) ... done in  0.0389 seconds.
Building Bia (ia|Q) ... done in  0.0244 seconds.
Computing DF-MP2 energy ... done in  0.1074 seconds.

Density fitted MP2 energy:  -0.2107758942
Total DF-MP2 energy:       -75.8540935937 (HF + DF-MP2)
DF-MP2 total time:  0.2743 seconds.

DF error:                   -0.0000041511
```