

Polyatomic Self-Consistent Field with Perturbative Energy Correction: Dynamical Electron Correlation

Coleman Howard

May 9, 2012

1 Introduction

In my last project, I demonstrated the Hartree-Fock (HF) approximation to the time-independent Schrödinger equation. HF theory is a starting point for many of the most successful *ab initio* methods commonly used in quantum chemistry research today. More accurate methods employ the HF wave function as a zeroth or first order approximation because the electronic energy computed within the approximation has less than 1% error. The major source of error in HF is due to the fact that electron repulsion is treated in an average sense. A truly accurate description of electrons would account for the correlated motions of the electrons in a dynamical sense, as electrons experience instantaneous mutual repulsion. Here, I will demonstrate how perturbation theory is used to recover some of this dynamical correlation. Also, this work will improve on the previous effort by considering atoms larger than hydrogen and helium, which only possess spherical electron probability distributions. Charge distributions in larger atoms are not strictly spherical, significantly complicating the calculation of the required molecular integrals.

The goal is to calculate the total energy of a single water molecule (H_2O). To recover some of the dynamical electron correlation, we will compute the second-order Møller-Plesset energy (MP2 energy.) Later, we will examine the form of the final expression to compute this energy without a lengthy derivation. For now, we will simply say that the MP2 energy is computed from the HF energy with a perturbative correction that includes contributions from the electron-repulsion integrals over unoccupied orbitals. The HF wave function is calculated in a self-consistent field (SCF) procedure, in the exact same manner as the previous work. However, the integrals required now involve six-dimensional integrals over basis functions representing non-spherical electron probability distributions. Let's begin by examining the mathematical form of the basis functions and the kinds of terms that result when trying to compute the four different types of integrals required in the SCF procedure.

The basis function utilized here is the STO-3G basis set, referred to as a minimal basis set because it contains only one basis function for each occupied orbital for each atom. For example, the hydrogen atom has one electron, which is said to reside in an *s* orbital, where the probability distribution is spherically symmetric. The helium atom has two electrons which are said to reside in an *s* orbital. Oxygen, with eight electrons, has two electrons occupying an *s* orbital (1s), two more electrons occupying a different *s* orbital (2s), and four electrons distributed among what are referred to as *p* orbitals (2p). The labels *s* and *p* actually come from spectroscopic terminology in reference to characteristic lines from atomic spectra: *s* for sharp and *p* for principal. The 3-dimensional pictorial representation of a *p* orbital is commonly shown in undergraduate chemistry textbooks as having a dumbbell shape.

The STO-3G basis set for the hydrogen atom takes the following form:

$$\phi_{1s} = \sum_{i=1}^3 d_{i,1s} g_{1s}(\alpha_{i,1s})$$
$$g_{1s}(\alpha_{i,1s}) = N \exp[-\alpha_i r^2]$$

In the first expression, the contraction coefficients d_i and exponents α_i are predefined, and

have been chosen to closely approximate the canonical wave function solutions to the hydrogen atom Schrödinger equation. The second expression shows the form of a normalized primitive Gaussian, which when combined in the contracted series of the first expression, defines a single basis function. Similarly, the STO-3G basis set for the oxygen atom is defined below:

$$\begin{aligned}
\phi_{1s} &= \sum_{i=1}^3 d_{i,1s} g_{1s}(\alpha_{i,1s}) \\
\phi_{2s} &= \sum_{i=1}^3 d_{i,2s} g_{1s}(\alpha_{i,2sp}) \\
\phi_{2p_x} &= \sum_{i=1}^3 d_{i,2p_x} g_{2p}(\alpha_{i,2sp}) \\
\phi_{2p_y} &= \sum_{i=1}^3 d_{i,2p_y} g_{2p}(\alpha_{i,2sp}) \\
\phi_{2p_z} &= \sum_{i=1}^3 d_{i,2p_z} g_{2p}(\alpha_{i,2sp}) \\
g_{2p_x}(\alpha_{i,1s}) &= N \ x \ \exp[-\alpha_i r^2]
\end{aligned}$$

The subscripts above are meaningful and indicate that the contraction coefficients are the same for all 1s functions, regardless of the atom type. Notice also that the gaussian form of the 1s function will be the same as the 2s function, with only the exponents changing. The last expression above shows the Gaussian functional form for the 2p_x function. The 2p_y and 2p_z functions take analogous forms. So with the 5 basis functions for the oxygen atom and 1 basis function for each of the hydrogen atoms, we are ready to look at the molecular integrals over these functions.

2 Molecular Integral Evaluation

There are four types of molecular integrals that need to be computed: overlap integrals, kinetic energy integrals, nuclear attraction integrals, and electron-repulsion integrals. The first three categories are one-electron integrals over the x,y and z Cartesian components of space. Each of these types of integrals consist of a basis function on the left-hand side of the integral, an operator in the middle, and another basis function on the right-hand side. We defined a basis function earlier as being a linear combination of contracted primitive Gaussian functions. So each integral involving two basis functions can be broken down into a sum of integrals over the two basis functions' component primitive Gaussian functions. The problem is more straightforward when cast in this manner, and it allows us to ignore the corresponding contraction coefficients while we evaluate the integral over two primitive Gaussians.

2.1 Overlap Integrals

The first class of integrals we need to evaluate is the overlap integral. Here, the operator in between the two basis functions is simply unity. The overlap integral between Gaussian primitives 1 and 2 (S_{12}) is given by the expression below:

$$S_{12} = \int g_1(\alpha_1, \mathbf{A}, l_1, m_1, n_1) g_2(\alpha_2, \mathbf{B}, l_2, m_2, n_2) d\mathbf{r}$$

Here, \mathbf{A} and \mathbf{B} represent the 3-dimensional spatial components of the two primitives' centers (i.e., the centers of the atoms which the functions represent.) Application of the Gaussian Product Theorem leads to a manageable expression for the overlap:

$$\begin{aligned} S_{12} &= \exp[-\alpha_1\alpha_2(\overline{AB})^2/\gamma] I_x I_y I_z \\ \gamma &= \alpha_1 + \alpha_2 \\ I_x &= \sum_{i=0}^{l_1+l_2} f_i(l_1, l_2, \overline{PA}_x, \overline{PB}_x) \int_{-\infty}^{\infty} x_P^i \exp[-\gamma x_P^2] \\ &= \sum_{i=0}^{l_1+l_2} f_i(l_1, l_2, \overline{PA}_x, \overline{PB}_x) \frac{(2i-1)!!}{(2\gamma)^i} \left(\frac{\pi}{\gamma}\right)^{1/2} \\ f_k &= \sum_{q=\max(-k, k-2l_2)}^{\min(k, 2l_1-k)} \binom{l_1}{i} \binom{l_2}{j} (\overline{\mathbf{PA}})_x^{l_1-i} (\overline{\mathbf{PB}})_x^{l_2-j} \end{aligned}$$

In the preceding expressions and in future ones, l_1 and l_2 refer to angular momentum vector components in the x-direction associated with a particular primitive Gaussian. For our purposes, l will equal 1 for a p_x function and 0 in all other cases. Likewise, the letters m and n serve the same purpose for angular momentum in the y and z directions, respectively. The letter \mathbf{P} will be reserved for the center of a new Gaussian function formed from the two primitive Gaussians by using the Gaussian Product theorem.

From a programming standpoint, the overlap integral over two primitive Gaussians is not particularly daunting. The expression above is actually general for Gaussian basis functions of arbitrary angular momentum, not just s and p functions. The only piece of the expressions above that I have not coded before is the double factorial. However, the code is just as simple as the single factorial, except a multiplier value is decremented by 2 each time, instead of 1. The function I wrote to compute the overlap includes access to factorial, double factorial, and binomial coefficient functions. The information required by the function is all present in attributes associated with each basis function. So the main function sends two basis set objects to the overlap function, along with two numbers specifying which primitives are to be used. These numbers allow the overlap function to retrieve information such as α values and angular momentum components needed to perform the calculation. The contraction coefficients are then applied to the result inside the main function.

2.2 Kinetic Energy Integrals

The operator of interest for the kinetic energy integrals is the Cartesian Laplacian operator. Conveniently, the kinetic energy integral between 2 primitive Gaussians (T_{12}) can be reduced to a series of overlap integrals:

$$\begin{aligned} T_{12} &= I_x + I_y + I_z \\ I_x &= \frac{1}{2} l_1 l_2 \langle -1 | -1 \rangle_x + 2\alpha_1 \alpha_2 \langle +1 | +1 \rangle_x \\ &\quad - \alpha_1 l_2 \langle +1 | -1 \rangle_x - \alpha_2 l_2 \langle -1 | +1 \rangle_x \\ \langle +1 |_x &= x^{l_1+1} y^{m_1} z^{n_1} \exp[-\alpha_1 r^2] \end{aligned}$$

So the kinetic energy integrals can be generated completely from a set of associated overlap integrals with angular momentum quantum numbers incremented or decremented by one unit.

2.3 Nuclear-Attraction Integrals

The operator inside the nuclear-attraction integrals is \mathbf{R}_C^{-1} , the displacement vector from a basis function to a nuclear center. Unfortunately, with the introduction of the nuclear-attraction integral, we have encountered an integral which is not separable due to the nature of the operator. A transformation is required, and there are multiple equivalent techniques. The method I have chosen for this work is using recurrence relations. This was chosen simply because it looked like the easiest to code. It is based on the method of Obara and Saika, who showed how to arrive at the solution to the Cartesian integral without having to compute intermediate Hermite integrals as in other methods. The method is best described by the concept of a source integral and a target integral:

$$\begin{aligned} \Theta_{000000}^N &= \frac{2\pi}{\gamma} K_{ab}^{xyz} F_N(\alpha R_{PC}^2) \\ \Theta_{ijklmn}^0 &= V_{ijklmn}^{000} \end{aligned}$$

The first expression is the source integral, meaning the starting point. I can set up a function to compute this expression. The second expression, the target integral, refers to the particular integral of interest. The goal is to use recursion relations to express the target integral in terms of the known source integral. In the source integral, the first two indices i and j refer to the values of l_1 and l_2 (i.e., the angular momentum quantum numbers of the primitive Gaussian functions in the x-direction.) The remaining four lower indices refer to the two Gaussians' y- and z-components of quantized angular momentum. We will now define the crucial recursion relation:

$$\begin{aligned} \Theta_{i+1,j,k,l,m,n}^N &= X_{PA} \Theta_{ijklmn}^N + \frac{1}{2\gamma} (i \Theta_{i-1,j,k,l,m,n}^N + j \Theta_{i,j-1,k,l,m,n}^N) \\ &\quad - X_{PC} \Theta_{ijklmn}^{N+1} - \frac{1}{2\gamma} (i \Theta_{i-1,j,k,l,m,n}^{N+1} + j \Theta_{i,j-1,k,l,m,n}^{N+1}) \end{aligned}$$

Analogous expressions are obtained for incrementing or decrementing the other indices. But let us look at the evaluation of the source integral in some detail. In the source integral, the key to evaluation is the approximation to the Boys function (F_N). The preceding factors are easily calculated from the defined qualities of the basis functions. The Boys function of order N is defined in the following manner:

$$F_N(x) = \int_0^1 \exp[-xt^2] t^{2N} dx$$

$$F_N(0) = \frac{1}{2n+1}$$

$$F_N(x) \approx \frac{(2n-1)!!}{2^{n+1}} \sqrt{\frac{\pi}{x^{2n+1}}} \quad (\text{large } x)$$

The Boys functions is actually the only numerical approximation that must be made in this program. It is also used in the computation of the electron-repulsion integrals so an accurate approximation is essential. The strategy used in this program is that suggested by Helgaker, et al. The domain of the Boys function is divided into 3 regions: (i) $x < 0.18$, (ii) $0.18 < x < 19.35$ and (iii) $x > 19.35$. For the first region, a sixth-order Taylor series expansion is stable, resulting in greater than 10 decimal points of precision. Let's examine the Python code for this series:

```
def TAYLOR(x,order,sub):
    total=0.0
    for k in range(order+1):
        total+=((-1*x)**k)/factorial(k)/(2*sub+2*k+1)
    return total
```

Here, the variable *sub* refers to the order of the Boys function, while *order* refers to the order of the Taylor series expansion, which is chosen as 6 in all cases. In the third region of the Boys domain (large x), the third expression above can be used. Now, the most difficult region is the middle region where neither a Taylor expansion nor the large x expression is stable. The solution recommended by Helgaker is to tabulate the Boys function values from 0.18 to 18.35 in increments of 0.1, and so the approximation in this region can be found by expanding around the nearest tabulated point. This strategy leads to convergence with errors smaller than 10^{-14} . The values were tabulated using Mathematica software for Boys function orders 1 through 8. These values were placed into 8 separate dictionaries in one file, mapping an x value to the corresponding Mathematica result. Here is the function:

```
def BOYS(x,N,order=6):
    total=0.0
    xten=x*10
    closest=round(xten)/10
    diff=x-closest
```

```

for k in range(order+1):
    marker=k+N
    if marker==1:
        mydict=boysONE
    if marker==2:
        mydict=boysTWO
    if marker==3:
        mydict=boysTHREE
    if marker==4:
        mydict=boysFOUR
    if marker==5:
        mydict=boysFIVE
    if marker==6:
        mydict=boysSIX
    if marker==7:
        mydict=boysSEVEN
    if marker==8:
        mydict=boysEIGHT
    total+=mydict[closest]*(-1*diff)**k/factorial(k)
return total

```

So with a stable approximation to Boys function, computing the nuclear attraction integrals is as simple as deriving the target integral in terms of the Boys function source integral. At this point, I would like to explain my ingenious algorithm for deriving the recursion relations, but I don't have one. And I doubt it would enhance the program if I did. Because this computation is only dealing with a maximum angular momentum quantum of 1, there is no clear advantage to coding a smart algorithm rather than deriving the few needed equations by hand. In addition, to extend this program to the d or f functions of angular momenta 2 and 3, respectively, hard coding would still probably be the preferred method. In my research, I have never used a function with angular momenta higher than 3.

2.4 Electron Repulsion Integrals

The last set of integrals is defined by the operator \mathbf{r}_{12}^{-1} , representing the distance between electrons. For the electron-repulsion integrals (ERI), the recurrence relations of Obara and Saika are called upon once again. The source and target integrals are defined below:

$$\Theta_{0000;0000;0000}^N = \frac{2\pi^{5/2}}{pq\sqrt{p+q}} K_{ab}^{xyz} K_{cd}^{xyz} F_N(\alpha R_{PQ}^2)$$

$$\Theta_{i_x j_x k_x; j_y k_y l_y; j_z k_z l_z}^0 = g_{i_x j_x k_x; j_y k_y l_y; j_z k_z l_z}$$

So there are 12 lower indices in all, 3 angular momentum quantum numbers for each of the four primitive Gaussians. Here, p has replaced γ and q is the corresponding value for the

Gaussian product of the two primitives on the right hand side of the integral. Luckily the recurrence relations for the x-components are not affected by those of the y- or z-components, and vice versa. Let's examine the form of the Obara-Saika two-electron recurrence relation in the x direction. We will drop the unaffected indices.

$$\begin{aligned}\Theta_{i+1,j,k,l}^N = & X_{PA}\Theta_{ijkl}^N - \frac{\alpha}{p}X_{PQ}\Theta_{ijkl}^{N+1} + \frac{i}{2p}\left(\Theta_{i-1,j,k,l}^N - \frac{\alpha}{p}\Theta_{i-1,j,k,l}^{N+1}\right) \\ & + \frac{j}{2p}\left(\Theta_{i,j-1,k,l}^N - \frac{\alpha}{p}\Theta_{i,j-1,k,l}^{N+1}\right) + \frac{k}{2(p+q)}\Theta_{i,j,k-1,l}^{N+1} + \frac{l}{2(p+q)}\Theta_{i,j,k,l-1}^{N+1}\end{aligned}$$

Again, the programmer is left with some equations to derive, but all of the information needed is conveniently present inside the basis function object from the class `Gaussian3D`. And the concept needed to compute the source integral (lower indices all equal to zero) has already been used to compute the nuclear attraction integrals.

3 Perturbative Energy Correction

MP2 has been recognized as an efficient and relatively simple method of recovering dynamical electron correlation missing from the HF energy. The method uses the so-called “canonical orbitals” from the SCF procedure. The procedure for obtaining these has been explained in the past effort. Now to obtain the MP2 energy only two steps are required: (i) a transformation of the atomic orbital (AO) basis into a molecular orbital (MO) basis and (ii) the calculation of the energy. The basis transformation need only be applied to the matrix containing the results of the ERI. A loop similar to the following accomplishes our goal:

```
for p in xrange(nbasis):
    for q in xrange(nbasis):
        for r in xrange(nbasis):
            for s in xrange(nbasis):
                val=0.0
                for i in xrange(len(C)):
                    for j in xrange(len(C)):
                        for k in xrange(len(C)):
                            for l in xrange(len(C)):
                                val+=C[i][p]*C[j][q]*ERI[i][j][k][l]*C[k][r]*C[l][s]
                newERI[p][q][r][s]+=val
```

This 8-layer loop is responsible for the larger scaling of the MP2 method. The above algorithm is the naive method, as smarter loops can scale as N^5 , but this exercise is not for achieving efficiency. After transforming the ERI into the MO basis (`newERI`), we have everything needed to compute the MP2 energy:


```

emp2=0.0
for i in range(ndocc):
    for a in range(ndocc,nbasis):
        for j in range(ndocc):
            for b in range(ndocc,nbasis):
                emp2+=newERI[i][a][j][b]*(2*newERI[i][a][j][b]-newERI[i][b][j][a])
                / (E[i][i]+E[j][j]-E[a][a]-E[b][b])

```

So emp2 represents the dynamical electron correlation missing from the HF energy computation. In the above expression, the E values referenced are those from the SCF procedure obtained by diagonalizing the Fock matrix. ndocc is the number of doubly-occupied orbitals in the molecular system. If we assume all electrons are paired, then ndocc equals half the number of electrons.

4 Conclusions

The main result of the program is that it works. The integrals computed match those of **Gaussian09** to as many decimal places as **Gaussian09** will print. Also the MP2 energy matches to as many decimal places as I want. The number of decimal places that can match on the MP2 energy is limited to the convergence of the Hartree-Fock wave function. For this program, I have set an extremely tight SCF convergence criterion of a root mean square (RMS) deviation in the density matrix between successive iterations of less than 1×10^{-17} . Default convergence in **Gaussian09** is 1×10^{-8} . So this program's convergence is very tight. The greatest difficulty in this program was computing the electron-repulsion integrals. The code is both unattractive and inefficient, but it does produce accurate results. Currently, the best algorithms for computing these integrals involve starting with a tree-search routine for finding the most efficient way to only compute the necessary integrals. A result of using recurrence relations to compute integrals is that the repulsion integrals associated with lower-order angular momentum quantum numbers must be computed to obtain the higher-order results. On one hand, this could be advantageous if the results were stored and recalled efficiently since they will be used often. On the other hand, this means that recurrence relations could require some extra floating point operations which may exceed the demand of other computational schemes.

5 References

All of the information here is from *Molecular Electronic-Structure Theory*, T. Helgaker, P. J Jørgensen, and J. Olsen, Wiley, **2000**.