Joshua Goings

MP2 with Python
15 Jun 2013

If you have the transformed two electron integrals after a
Hartree-Fock calculation, it's trivial to perform Moller-
Plesset Second Order Perturbation theory (MP2). It is a
non-iterative way to calculate electron correlation energy
(e.g. the energy obtained from having electrons "avoid"
each other in molecules). The expression for MP2 energy
contribution (in a *spin orbital* basis) looks like this:

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

Where i and j are the indices of the occupied orbitals, and
a and b are the indices of the virtual orbitals. The
numerator is a "double-bar integral", which accounts for
the Coulomb and Exchange energies between pairs of
electrons, and the bottom are the orbital energies
(eigenvalues). The numerator is a result of transforming
the two-electron integrals to a molecular orbital basis,
and the orbital energies are the result of a Hartree-Fock
calculation. (Well, strictly speaking *all* of this is from a
Hartree-Fock calculation...) You can find a derivation of
this in Szabo and Ostlund (1996).

This is actually one of the easiest electronic structure
methods to program. It is non-iterative (which means no
guess vectors, chaotic behavior, etc), and can be readily
constructed from integrals on hand.

Computationally speaking, we can see that we must
construct four loops, to loop over our four indices. The
first two must loop over occupied orbitals, and the second
two must loop over virtual orbitals. I've attached some
Python code that accomplishes this. Similar to the CIS and
TDHF code, I have hard coded the transformed integrals
for HeH+ in an STO-3G basis, at a bond length of 0.9295
Angstroms.

```python
#!/usr/bin/python

####################################
#
#  Moller-Plesset Second Order
#   Perturbation Theory (MP2)
#
####################################

from __future__ import division
import math
import numpy as np

####################################
#
#    FUNCTIONS
#
####################################

# Return compound index given four indices
def eint(a,b,c,d):
    if a > b: ab = a*(a+1)/2 + b
    else: ab = b*(b+1)/2 + a
    if c > d: cd = c*(c+1)/2 + d
    else: cd = d*(d+1)/2 + c
    if ab > cd: abcd = ab*(ab+1)/2 + cd
    else: abcd = cd*(cd+1)/2 + ab
    return abcd

# Return Value of spatial MO two electron integral
# Example: (12\vert 34) = tei(1,2,3,4)
def teimo(a,b,c,d):
    return ttmo.get(eint(a,b,c,d),0.0e0)

####################################
#
#   INITIALIZE ORBITAL ENERGIES
#   AND TRANSFORMED TWO ELECTRON
#   INTEGRALS
#
####################################

Nelec = 2 # we have 2 electrons in HeH+
dim = 2 # we have two spatial basis functions in STO-
E = [-1.52378656, -0.26763148]
ttmo = {5.0: 0.94542695583037617, 12.0: 0.17535895381

####################################################
#
#   CONVERT SPATIAL TO SPIN ORBITAL MO
#
####################################################

# This makes the spin basis double bar integral (phys
# We double the dimension (each spatial orbital is nc
dim = dim*2
ints=np.zeros((dim,dim,dim,dim))
for p in range(1,dim+1):
    for q in range(1,dim+1):
        for r in range(1,dim+1):
            for s in range(1,dim+1):
```

```
            value1 = teimo((p+1)//2,(r+1)//2,(q+1
            value2 = teimo((p+1)//2,(s+1)//2,(q+1
            ints[p-1,q-1,r-1,s-1] = value1 - valu


####################################################
#
#   Spin basis fock matrix eigenvalues
#
####################################################

fs = np.zeros((dim))
for i in range(0,dim):
     fs[i] = E[i//2]
   #fs = np.diag(fs)


####################################################
#
#   MP2 Calculation
#
####################################################

# First two loops sum over occupied spin orbitals
# which is equal to the number of electrons. The last
# two loops loop over the virtual orbitals.
EMP2 = 0.0
for i in range(0,Nelec):
    for j in range(0,Nelec):
        for a in range(Nelec,dim):
            for b in range(Nelec,dim):
                EMP2 += 0.25*ints[i,j,a,b]*ints[i,j,a

print "E(MP2) Correlation Energy = ", EMP2, " Hartree
```

If you run the code you'll get a correlation energy of -0.00640 Hartrees. This is pretty small, and in fact most correlation energies are. Hartree-Fock does a good job at recovering ~99% of the energy of a molecule. However, correlation energies still translate to the order of 10 kcal/mol, which is chemically significant. This is why including correlation effects are so important. It's hard to make accurate predictions about reactivities with such large error bars when you neglect correlation. MP2 is an easy way to do so: much more accurate methods, like coupled-cluster methods, will give you energetics to chemical accuracy (chemical accuracy is ~0.1 kcal/mol). However, they are often very costly!

Related Posts

The Magnus Expansion 15 Jun 2017

A (hopefully) gentle guide to the computer implementation of molecular integrals over Gaussian basis functions. 28 Apr 2017

Quantum Dynamics Movies 20 Apr 2016