

Instructor: Prof. Lee M. Thompson E-mail: *lee.thompson.1@louisville.edu* Office: CB 251

1. Using a two electron system, demonstrate that the Hartree product wavefunction (product of one-electron wavefunctions) is not antisymmetric, in contrast to the Slater determinant.

$$|\psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2)\rangle \neq |\psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2)\rangle$$

$$|\psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) - \psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2)\rangle = -|\psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2) - \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2)\rangle$$

2. Derive the normalization constant of the Slater determinant – $(1/n!)^{1/2}$

$$|\Psi_{SD}\rangle = N \sum_{\pi \in S_n} \epsilon_{\pi} \hat{\pi} |\psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots\rangle$$

$$N^2 \sum_{\pi \in S_n} \sum_{\pi' \in S_n} \epsilon_{\pi} \epsilon_{\pi'} \langle \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots | \hat{\pi}^{\dagger} \pi | \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots \rangle = 1$$

$$N^2 \{ \langle \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots | \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots \rangle - \langle \psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2) \dots | \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots \rangle$$

$$- \langle \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2) \dots | \psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2) \dots \rangle + \langle \psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2) \dots | \psi_2(\mathbf{r}_1)\psi_1(\mathbf{r}_2) \dots \rangle + \dots \} = 1$$

$$N^2 \{1 - 0 - 0 + 1 \dots\} = 1$$

so the only terms that survive are when the left and right permutation are the same. There are $n!$ such terms so the sum in braces must be $n!$

$$N^2 \{n!\} = 1$$

$$N = (1/n!)^{1/2}$$

3. Show that the Hartree product wavefunction is an eigenfunction of the non-interacting system of electrons and determine the corresponding eigenvalue.

$$\hat{H} = \sum_{i=1}^{N_{elec}} \hat{h}_i$$

$$|\Psi_{HP}\rangle = |\psi_1\psi_2\psi_3 \dots\rangle$$

$$\hat{H}|\Psi_{HP}\rangle = \sum_{i=1}^{N_{elec}} \hat{h}_i |\psi_1\psi_2\psi_3 \dots\rangle$$

$$\hat{H}|\Psi_{HP}\rangle = \sum_{i=1}^{N_{elec}} \epsilon_i |\psi_1\psi_2\psi_3 \dots\rangle$$

$$E = \sum_{i=1}^{N_{elec}} \epsilon_i$$

4. Show that the wavefunction described by the Slater determinant $|\Psi_{SD}\rangle = (1/6)^{1/2} |\psi_1\psi_2\psi_1\rangle$ is zero.

$$|\Psi_{SD}\rangle = (1/6)^{1/2} |\psi_1\psi_2\psi_1\rangle$$

$$|\Psi_{SD}\rangle = (1/6)^{1/2} \{ \psi_1\psi_2\psi_1 - \psi_2\psi_1\psi_1 - \psi_1\psi_2\psi_1 - \psi_1\psi_1\psi_2 + \psi_2\psi_1\psi_1 + \psi_1\psi_1\psi_2 \} = 0$$

5. In lectures, we have studied the Slater determinant and how the energy of a Slater determinant wavefunction can be computed, assuming the form of the molecular orbitals are known. In this worksheet, we will extract the one and two electron integrals (h_{ij} , J_{ij} and K_{ij}) computed using molecular orbitals from a Gaussian calculation, and use them to compute the energy of the Slater determinant wavefunction. Copy the following text into a new .f03 file (e.g. SDenergy.f03):

```
1  program SDenergy
2  !
3  use mqc_gaussian
4  use iso_fortran_env
5  !
```

```

6 !      Variable Declarations ...
7 !
8     implicit none
9     character(len=:), allocatable :: fileName
10    type(mqc_gaussian_unformatted_matrix_file):: fileInfo
11    integer(kind=int64)::iOut=6,iPrint=2,IAlpha,IBeta,JAlpha,JBeta,NAlpha,NBeta
12    logical :: UHF
13    type(mqc_pscf_wavefunction)::wavefunction
14    type(mqc_molecule_data)::moleculeInfo
15    type(mqc_twoERIs)::eris,mo_ERIs
16    type(mqc_scf_integral)::mo_core_ham
17    type(mqc_scalar)::Vnn,alpha_one_energy,beta_one_energy,alpha_two_energy_ss,&
18        beta_two_energy_ss,two_energy_os,final_energy
19 !
20 !      Get the user-defined filename from the command line and then call the
21 !      routine that reads the Gaussian matrix file.
22 !
23    call mqc_get_command_argument(1,fileName)
24    call fileInfo %load(filename)
25    call fileInfo %getESTObj('wavefunction',wavefunction)
26    call fileInfo %getMolData(moleculeInfo)
27    call fileInfo %get2ERIs('regular',eris)
28 !
29 !      Determine if the wavefunction that we read in was restricted or unrestricted.
30 !
31    if (wavefunction%wf_type.eq.'U') then
32        UHF = .true.
33        write(*,*) 'Found unrestricted wavefunction'
34    elseif (wavefunction%wf_type.eq.'R') then
35        UHF = .false.
36        write(*,*) 'Found restricted wavefunction'
37    else
38        call mqc_error('Unsupported wavefunction type')
39    endif
40    if (wavefunction%wf_complex) call mqc_error('Unsupported wavefunction type')
41    NAlpha = wavefunction%NAlpha
42    NBeta = wavefunction%NBeta
43 !
44 !      Compute the nuclear-nuclear repulsion energy.
45 !
46    Vnn = mqc_get_nuclear_repulsion(moleculeInfo)
47    call Vnn%print(iOut,'Nuclear Repulsion Energy (au)')
48 !
49 !      Transform one and two-electron integrals to MO basis. These are the hij, Jij and Kij terms.
50 !
51    if (iPrint.eq.1) write(iOut,*) 'Transforming MO integrals'

```

```

52 mo_core_ham = matmul(transpose(wavefunction%MO_Coefficients),&
53     matmul(wavefunction%core_Hamiltonian, &
54         Wavefunction%MO_Coefficients))
55 if (IPrint.ge.2) call mo_core_ham%print(iOut,'MO Basis Core Hamiltonian')
56 call twoERI_trans(iOut,iPrint,wavefunction%MO_Coefficients,ERIs,mo_ERIs)
57 !
58 !     Compute Slater determinant energy.
59 !     For restricted , E = 2*sum_{ij}hii + 2*sum_{ij}Jij - sum_{ij}Kij
60 !     For unrestricted , E = sum_{i}hiaia + sum_{i}hibib + sum_{ij}Jiaja + sum_{ij}Jibjb +
61 !         2*sum_{ij}Jiajb - sum_{ij}Kiaja - sum_{ij}Kibjb
62 !
63 !     one-electron term
64 alpha_one_energy = 0.0
65 do IAlpha = 1, NAlpha
66     alpha_one_energy = alpha_one_energy + mo_core_ham%at(IAlpha,IAlpha,'alpha')
67 endDo
68 if (.not.UHF) then
69     alpha_one_energy = 2*alpha_one_energy
70 else
71     beta_one_energy = 0.0
72     do IBeta = 1, NBeta
73         beta_one_energy = beta_one_energy + mo_core_ham%at(IBeta,IBeta,'beta')
74     endDo
75 endIf
76 call alpha_one_energy%print(iOut,'Alpha one electron energy: haa')
77 If(UHF) call beta_one_energy%print(iOut,'Beta one electron energy: hbb')
78 !
79 !     two-electron same-spin term
80 alpha_two_energy_ss = 0.0
81 do IAlpha = 1, NAlpha
82     do JAlpha = IAlpha+1, NAlpha
83         alpha_two_energy_ss = alpha_two_energy_ss + &
84             mo_ERIs%at(IAlpha,IAlpha,JAlpha,JAlpha,'alpha') - &
85             mo_ERIs%at(IAlpha,JAlpha,JAlpha,IAlpha,'alpha')
86     endDo
87 endDo
88 if (.not.UHF) then
89     alpha_two_energy_ss = 2*alpha_two_energy_ss
90 else
91     beta_two_energy_ss = 0.0
92     do IBeta = 1, NBeta
93         do JBeta = IBeta+1, NBeta
94             beta_two_energy_ss = beta_two_energy_ss + &
95                 mo_ERIs%at(IBeta,IBeta,JBeta,JBeta,'beta') - &
96                 mo_ERIs%at(IBeta,JBeta,JBeta,IBeta,'beta')
97         endDo

```

```

98     endDo
99   endIf
100   call alpha_two_energy_ss%print(iOut,'Alpha same spin two electron energy: Jaa - Kaa')
101   if (UHF) call beta_two_energy_ss%print(iOut,'Beta same spin two electron energy: Jbb - Kbb')
102 !
103 !   two-electron opposite-spin term
104   two_energy_os = 0.0
105   do IAlpha = 1, NAlpha
106     do IBeta = 1, NBeta
107       if (UHF) then
108         two_energy_os = two_energy_os + &
109         mo_ERIs%at(IAlpha,IAlpha,IBeta,IBeta,'alphaBeta')
110       else
111         two_energy_os = two_energy_os + &
112         mo_ERIs%at(IAlpha,IAlpha,IBeta,IBeta,'alpha')
113       endif
114     endDo
115   endDo
116   call two_energy_os%print(iOut,'Opposite spin two electron energy (Coulomb only): Jab')
117 !
118   if (.not.UHF) then
119     final_energy = alpha_one_energy + alpha_two_energy_ss + two_energy_os
120   else
121     final_energy = alpha_one_energy + beta_one_energy + alpha_two_energy_ss + &
122     beta_two_energy_ss + two_energy_os
123   endif
124   call final_energy%print(iOut,'Total electronic energy')
125 !
126   final_energy = Vnn + final_energy
127   call final_energy%print(iOut,'Total electronic energy + nuclear repulsion energy')
128 !
129   End Program SDEnergy
130

```

Create a makefile for this code and compile the code. If you have errors, try to debug the code by understanding the part of the code that the compiler is complaining about. Once the code has compiled without error make a file named h2_rhf.com and add the following text:

```

1 #P rhf nosymm output=matrix int=noraf scf=conven
2
3 test
4
5 0 1
6 H

```

```
7 H 1 1.1
8
9 h2_rhf.mat
```

Run the file with 'g16 h2_rhf.com' and make sure the file completed without error and produced a matrix file h2_rhf.mat. Use the .mat file as input to your code by typing './SDenergy.exe h2_rhf.mat'. You should find that the energy you get out of your code is the same as the energy listed in the Gaussian h2_rhf.log file on the line containing 'SCF Done:'

Your code has taken the orbitals produced by the Gaussian calculation and used them to produce the energy of a Slater determinant wavefunction. Your code has broken the energy contributions into one-electron terms, same-spin two-electron terms, and opposite-spin two-electron terms. These three components make up the electronic energy, and combined with the nuclear repulsion energy give the overall energy. Note the same spin two-electron term is zero because in H_2 there is only one electron of each spin.

Make a new file named h2_uhf.com and add the following text:

```
1 #P uhf guess=mix stable=opt nosymm output=matrix int=noraf scf=conver
2
3 test
4
5 0 1
6 H
7 H 1 1.1
8
9 h2_uhf.mat
```

This file is the same as the previous h2_rhf.com file but produces an unrestricted calculation. Execute this input file with Gaussian to produce the matrix file and use this as input to your SDenergy.exe code. Note that the energy of the restricted and unrestricted calculations are the same. The reason unrestricted does not lower the energy is because around the equilibrium geometry, the 1s orbitals on each hydrogen atom interact strongly to make the σ bonding orbital much lower in energy than the σ^* antibonding orbital. As a result, there is a strong drive to pair orbitals in the same molecular orbital, and restricted orbitals are in fact the lowest energy orbitals possible. Now increase the bond length in both files from 1.1 Å to 5.0 Å in both h2_rhf.com and h2_uhf.com and rerun. Now use the new matrix files as input to your SDenergy code. Now you will find the unrestricted wavefunction is lower in energy than the restricted energy because, as the hydrogen atoms are far apart, the 1s orbitals interact weakly and the σ bonding orbital is not significantly lower in energy than the σ^* orbital. In the restricted picture, the double occupation of the σ bonding orbital leads to an ionic electron configuration in which both electrons are on a single hydrogen atom, while in the unrestricted picture, a single electron can reside on each hydrogen atom, so that homolytic cleavage can be correctly modeled.

Take a look through your code and compare the code to equations in the lecture notes. Can you identify the terms in the equations in the code and understand how the Slater determinant energy is computed? Try making matrix files for larger molecules or non-singlet multiplicities to see how the energy contributions change.