

ELECTRONIC STRUCTURE - PRACTICAL SESSION 2

Electronic structure for the molecular H_2^+ ion

Gisela Martí Guerrero

January 2023

1 Problem 4

- (a) Write a program to calculate the energy for the ground state and the first excited state of the H_2^+ molecular ion at a fixed value of the intranuclear distance R using an LCAO trial function made from two $1s$ STO with exponent k , one on each nucleus.

The ground state energy and the first excited state energy correspond to the bonding and the anti-bonding molecular orbitals of the H_2^+ ion. Their energies are defined by:

$$E(\text{ground state}) = \frac{H_{aa} + H_{ab}}{1 + S_{ab}}$$

$$E(\text{excited state}) = \frac{H_{aa} - H_{ab}}{1 - S_{ab}}$$

where

$$H_{aa} = \frac{k^2}{2} - k - \frac{1}{R} + \exp(-2kR) \left(k + \frac{1}{R} \right)$$

$$H_{ab} = -\frac{1}{2}k^2 S_{ab} + k(k-2)(1+kR)\exp(-kR)$$

$$S_{ab} = \exp(-kR) \left[1 + kR + \frac{1}{3}k^2 R^2 \right]$$

```
import numpy as np
import matplotlib.pyplot as plt

def energies(k, R):
    Saa = Sbb = 1
    Sab = Sba = np.exp(-k*R) * (1+k*R+(1/3)*(k**2)*(R**2))
    Haa = Hbb = (1/2)*(k**2) - k - 1/R + np.exp(-2*k*R) * (k+1/R)
    Hab = Hba = -1/2 * (k**2) * Sab + k*(k-2)*(1+k*R)*np.exp(-k*R)
    w1 = (Haa + Hab)/(1 + Sab)
    w2 = (Haa - Hab)/(1 - Sab)
    return w1,w2

# Aproximation for fixed values of k = 1.5 and R = 1.5
print("The ground state energy is: ", energies(1.5, 1.5)[0], "a.u.")
print("The first excited state energy is: ", energies(1.5, 1.5)[1], "a.u.")
```

The output of the code is:

The ground state energy is: -1.2233339459063912 a.u.

The first excited state energy is: -0.3653669251687033 a.u.

- (b) Run the program for different values of k between 0.4 and 2.0 to find the optimal value of this parameter for a set of intermolecular distances between 0.8 and 2.0 Å (remember that all equations above are given in atomic units).

```

matrix_size = 100 #matrix size = 100x100, where columns are R and rows are k
bohr = 1.88972 # change units from angstroms to bohr radius
R = np.linspace(0.8*bohr,2.0*bohr,matrix_size)
k = np.linspace(0.4,2.0,matrix_size)
E = []
for i in range(matrix_size):
    for j in range(matrix_size):
        E.append(energies(k[i], R[j])[0])
energy_values = np.array(E)
k_index = np.where(energy_values == min(energy_values))
optimal_k_index = k_index[0] / matrix_size
optimal_k = k[int(optimal_k_index)]
print("The optimal k value is: ", optimal_k)

```

The output of the code is:

The optimal k value is: 1.3535353535353536

- (c) Calculate the potential energy surface for the two lowest states of the H_2^+ molecular ion by adding the $1/R$ nuclear repulsion term to their electronic energy and use your calculated ground state PES to determine the equilibrium geometry for the H_2^+ ion.

The potential energy can be calculated with

$$U(R) = E(R) + \frac{1}{R}$$

```

#Ground state energy PES
PES_ground = energies(optimal_k, R)[0] + (1/R)
#First excited state PES
PES_excited = energies(optimal_k, R)[1] + (1/R)

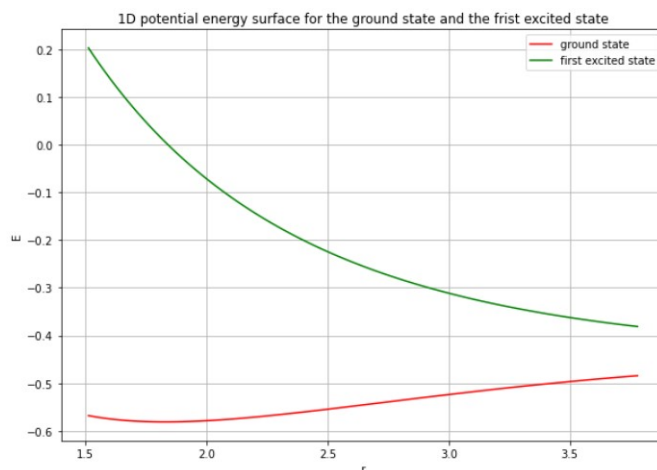
plt.figure(figsize=(9,6))
plt.title("1D potential energy surface for the ground state and the frist excited state")
plt.xlabel("r")
plt.ylabel("E")
plt.plot(R, PES_ground, label="ground state", color="red")
plt.plot(R, PES_excited, label="first excited state", color="green")
plt.legend()
plt.grid()

R_index = np.where(PES_ground == min(PES_ground))
R_equilibrium = R[int(R_index[0])]
print("The equilibrium distance for the H2+ molecule is: ",R_equilibrium, "bohr units")

```

The output of the code is:

The equilibrium distance for the H2+ molecule is: 1.8324557575757578 bohr units



2 Problem 5

Write a program to calculate the energy for the ground state and the first excited state of the H_2^+ molecular ion at a fixed value of the intranuclear distance R using an LCAO trial function made from two $1s$ STO-NG orbitals, one on each nucleus. Use the standard exponent $\zeta = 1.24$ in your program. Note: Your program should read the values of R and N from your input file.

The molecular orbitals' integrals are calculated with:

Overlap integrals

$$\langle A|B \rangle = \left[\frac{\pi}{(\alpha + \beta)} \right]^{3/2} \exp \left[-\frac{\alpha\beta}{(\alpha + \beta)} |R_A - R_B|^2 \right]$$

Kinetic energy integrals

$$\langle A | -\frac{1}{2} \nabla^2 | B \rangle = \frac{\alpha\beta}{\alpha + \beta} \left[3 - \frac{2\alpha\beta}{\alpha + \beta} |R_A - R_B|^2 \right] \times \left[\frac{\pi}{\alpha + \beta} \right]^{3/2} \exp \left[-\frac{\alpha\beta}{(\alpha + \beta)} |R_A - R_B|^2 \right]$$

Nuclear attraction integrals

$$\langle A | -\frac{Z_C}{r_{1C}} | B \rangle = -\frac{2\pi}{\alpha + \beta} Z_C \exp \left[-\frac{\alpha\beta}{(\alpha + \beta)} |R_A - R_B|^2 \right] \times F_0 [(\alpha + \beta) |R_P - R_C|^2]$$

```
import numpy as np
import scipy.special as sp

N = 3 #number of Gaussian primitives
R = 1.4 #intranuclear distance between the two atoms
zeta = 1.24
Za = Zb = 1.0
H = np.zeros([2,2])
S = np.zeros([2,2])
Rab2 = R**2

def S_int(A,B,Rab2): #A and B are the exponents of the atoms
    """
    Calculates the overlap between two gaussian functions
    """
    return (np.pi/(A+B))**1.5*np.exp(-A*B*Rab2/(A+B))

def T_int(A,B,Rab2):
    """
    Calculates the kinetic energy integrals for un-normalised primitives
    """
    return A*B/(A+B)*(3.0-2.0*A*B*Rab2/(A+B))*(np.pi/(A+B))**1.5*np.exp(-A*B*Rab2/(A+B))

def V_int(A,B,Rab2,Rcp2,Zc):
    """
    Calculates the un-normalised nuclear attraction integrals
    """
    V = 2.0*np.pi/(A+B)*F0((A+B)*Rcp2)*np.exp(-A*B*Rab2/(A+B))
    return -V*Zc

# Mathematical functions
def F0(t):
    """
    F function for 1s orbital
    """
    if (t<1e-6):
        return 1.0-t/3.0
    else:
        return 0.5*(np.pi/t)**0.5*sp.erf(t**0.5)
```

```

def erf(t):
    """
    Approximation for the error function
    """
    P = 0.3275911
    A = [0.254829592,-0.284496736,1.421413741,-1.453152027,1.061405429]
    T = 1.0/(1+P*t)
    Tn=T
    Poly = A[0]*Tn
    for i in range(1,5):
        Tn=Tn*T
        Poly=Poly*A[i]*Tn
    return 1.0-Poly*np.exp(-t*t)

Coeff = np.array([[1.00000,0.0000000,0.000000], #ST0-1G
                  [0.678914,0.430129,0.000000], #ST0-2G
                  [0.444635,0.535328,0.154329]]) #ST0-3G

Expon = np.array([[0.270950,0.000000,0.000000], #ST0-1G
                  [0.151623,0.851819,0.000000], #ST0-2G
                  [0.109818,0.405771,2.227660]]) #ST0-3G

D1 = np.zeros([3])
A1 = np.zeros([3])
D2 = np.zeros([3])
A2 = np.zeros([3])

# This loop constructs the contracted Gaussian functions
for i in range(N):
    A1[i] = Expon[N-1,i]*(zeta**2)
    D1[i] = Coeff[N-1,i]*((2.0*A1[i]/np.pi)**0.75)
    A2[i] = Expon[N-1,i]*(zeta**2)
    D2[i] = Coeff[N-1,i]*((2.0*A2[i]/np.pi)**0.75)

S12 = 0.0
T11 = 0.0
T12 = 0.0
T22 = 0.0
V11A = 0.0
V12A = 0.0
V22A = 0.0
V11B = 0.0
V12B = 0.0
V22B = 0.0

for i in range(N):
    for j in range(N):
        Rap = A2[j]*R/(A1[i]+A2[j])
        Rap2 = Rap**2 # Rap2 - squared distance between centre A and centre P
        Rbp2 = (R-Rap)**2
        S12 = S12 + S_int(A1[i],A2[j],Rap2)*D1[i]*D2[j]
        T11 = T11 + T_int(A1[i],A1[j],0.0)*D1[i]*D1[j]
        T12 = T12 + T_int(A1[i],A2[j],Rap2)*D1[i]*D2[j]
        T22 = T22 + T_int(A2[i],A2[j],0.0)*D2[i]*D2[j]
        V11A = V11A + V_int(A1[i],A1[j],0.0,0.0,Za)*D1[i]*D1[j]
        V12A = V12A + V_int(A1[i],A2[j],Rap2,Rap2,Za)*D1[i]*D2[j]
        V22A = V22A + V_int(A2[i],A2[j],0.0,Rap2,Za)*D2[i]*D2[j]
        V11B = V11B + V_int(A1[i],A1[j],0.0,Rap2,Zb)*D1[i]*D1[j]
        V12B = V12B + V_int(A1[i],A2[j],Rap2,Rbp2,Zb)*D1[i]*D2[j]
        V22B = V22B + V_int(A2[i],A2[j],0.0,0.0,Zb)*D2[i]*D2[j]

```

```

# Form core hamiltonian
H[0,0] = T11+V11A+V11B #Haa
H[0,1] = T12+V12A+V12B #Hab
H[1,0] = H[0,1]        #Hba
H[1,1] = T22+V22A+V22B #Hbb

# Form overlap matrix
S[0,0] = 1.0 #Saa
S[0,1] = S12 #Sab
S[1,0] = S12 #Sba
S[1,1] = 1.0 #Sbb

w1 = (H[0,0]+H[0,1]) / (1+S[0,1])
w2 = (H[0,0]-H[0,1]) / (1-S[0,1])
print("The ground state energy of the H2+ molecule is: ", w1, "a.u.")
print("The first excited state energy of the H2+ molecule is: ", w2, "a.u.")

```

The output of the code is:

The ground state energy of the H2+ molecule is: -1.2527982649241158 a.u.

The first excited state energy of the H2+ molecule is: -0.4756047975748145 a.u.