Untitled

August 16, 2021

[1]:
```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


get_ipython().run_line_magic('matplotlib', 'inline')

# In[2]:


import sympy as sp
from sympy import oo
import numpy as np
from itertools import product
from scipy.linalg import eig
from sympy import diff
import time as time
import matplotlib.pyplot as plt
from sympy.plotting import plot
# %matplotlib notebook
from IPython.display import Math

sp.init_printing()

r, r1, r2, r3, zeta, zeta1, zeta2 = sp.symbols("r, r1, r2,r3 zeta, zeta1,␣
↪zeta2")
n = sp.Symbol('n', integer=True)


def STO(zeta, n, r=r):
    return (2 * zeta) ** n * (2 * zeta / sp.factorial(2 * n)) ** (1 / 2) * r **␣
↪(n - 1) * sp.exp(-zeta * r)


# S Overlap Integrate
def S_int(f1, f2):
```

```
    return sp.integrate(f1 * f2 * r * r, (r, 0, +oo))


# H core = kinetics energy + electron and nuclear potential energy
def Hs_int(f1, f2, Z):
    return sp.integrate(f1 * (-((1 / 2) * (1 / r) * diff(diff(r * f2, r), r)) -␣
 ↪((Z / r) * f2)) * r * r, (r, 0, +oo))


# H core = kinetics energy + electron and nuclear potential energy
def Hp_int(f1, f2, Z):
    return sp.integrate(
        f1 * (-((1 / 2) * (1 / r) * diff(diff(r * f2, r), r)) + ((1 / r ** 2) *␣
 ↪f2) - ((Z / r) * f2)) * r * r,
        (r, 0, +oo))


# Returns the core hamiltonian matrix
def Hs_matrix(fs, Z):
    H = np.zeros((len(fs), len(fs)))
    for i in range(len(fs)):
        for j in range(len(fs)):
            H[i, j] = Hs_int(fs[i], fs[j], Z)

    return H


# Returns the overlap matrix
def S_matrix(fs):
    S = np.zeros((len(fs), len(fs)))
    for i in range(len(fs)):
        for j in range(len(fs)):
            S[i, j] = S_int(fs[i], fs[j])

    return S


# Returns the core hamiltonian matrix
def Hp_matrix(fp, Z):
    H = np.zeros((len(fp), len(fp)))
    for i in range(len(fp)):
        for j in range(len(fp)):
            H[i, j] = Hp_int(fp[i], fp[j], Z)

    return H
```

```
def Repulsion_electron(zetas, l):

    f1 = STO(zetas[0][0], zetas[0][1], r1)
    f2 = STO(zetas[1][0], zetas[1][1], r1)
    f3 = STO(zetas[2][0], zetas[2][1], r2)
    f4 = STO(zetas[3][0], zetas[3][1], r2)
    # f5 = STO(zetas[4][0], zetas[4][1], r3)
    # f6 = STO(zetas[5][0], zetas[5][1], r3)
    fs = [f1, f2, f3, f4]
    # need to fix var B and A
    B = (1 / r1 ** (l + 1)) * sp.integrate(f3 * f4 * r2 ** (l + 2), (r2, 0,␣
 ↪r1)) + r1 ** l * sp.integrate(
        f3 * f4 * r2 ** (1 - l), (r2, r1, +oo))
    A = sp.integrate(f1 * f2 * r1 * r1 * B, (r1, 0, +oo))
    return A


# Calculates Density matrix
# P need to be changed if the atom have unpaired electron
def Ps_matrix(Co):
    P = np.zeros([Co.shape[0], Co.shape[0]])

    for t in range(Co.shape[0]):
        for u in range(Co.shape[0]):
            P[t][u] = 2 * Co[t][0] * Co[u][0] + 2 * Co[t][1] * Co[u][1]
    return P


# Calculates Density matrix
# P need to be changed if the atom have unpaired electron
def Pp_matrix(Co):
    P = np.zeros([Co.shape[0], Co.shape[0]])

    for t in range(Co.shape[0]):
        for u in range(Co.shape[0]):
            P[t][u] = 2 * Co[t][0] * Co[u][0]
    return P


def J_matrix(zetai, zetaj, l):
    J = np.zeros((len(zetai), len(zetai), len(zetaj), len(zetaj)))

    rs = list(product(range(len(zetai)), repeat=2))
    tu = list(product(range(len(zetaj)), repeat=2))

    for r, s in rs:
        for t, u in tu:
```

```
            J[r, s, t, u] = Repulsion_electron((zetai[r], zetai[s], zetaj[t],␣
 ↪zetaj[u]), l)
    return J


def K_matrix(zetai, zetaj, l):
    K = np.zeros((len(zetai), len(zetai), len(zetaj), len(zetaj)))

    rs = list(product(range(len(zetai)), repeat=2))
    tu = list(product(range(len(zetaj)), repeat=2))

    for r, s in rs:
        for t, u in tu:
            K[r, s, t, u] = Repulsion_electron((zetai[r], zetaj[u], zetaj[t],␣
 ↪zetai[s]), l)
    return K


# Caculate G Matrix for 1s
def Gs_matrix(Cos, Cop, Jss, Kss, Jsp, Ksp):
    G = np.zeros((Cos.shape[0], Cos.shape[0]))
    Ps = Ps_matrix(Cos)
    Pp = Pp_matrix(Cop)

    rs = list(product(range(Cos.shape[0]), repeat=2))
    tu = list(product(range(Cos.shape[0]), repeat=2))
    ij = list(product(range(Cop.shape[0]), repeat=2))

    for r, s in rs:
        g = 0
        for t, u in tu:
            int1 = Jss[r, s, t, u]
            int2 = Kss[r, s, t, u]
            g += Ps[t, u] * (int1 - 0.5 * int2)
        for i, j in ij:
            int1 = Jsp[r, s, i, j]
            int2 = Ksp[r, s, i, j]
            g += Pp[i, j] * (int1 - 0.5 * int2)
        G[r, s] = g
    return G


# Returns the Fock matrix
def Fs_matrix(fs, Z, Cos, Cop, Jss, Kss, Jsp, Ksp):
    return Hs_matrix(fs, Z) + Gs_matrix(Cos, Cop, Jss, Kss, Jsp, Ksp)
```

```
# Caculate G Matrix for p
def Gp_matrix(Cos, Cop, Jpp, Kpp, Jps, Kps):
    G = np.zeros((Cop.shape[0], Cop.shape[0]))
    Ps = Ps_matrix(Cos)
    Pp = Pp_matrix(Cop)

    rs = list(product(range(Cop.shape[0]), repeat=2))
    tu = list(product(range(Cop.shape[0]), repeat=2))
    ij = list(product(range(Cos.shape[0]), repeat=2))

    for r, s in rs:
        g = 0
        for t, u in tu:
            int1 = Jpp[r, s, t, u]
            int2 = Kpp[r, s, t, u]
            g += Pp[t, u] * 0.5 * (int1 - int2)
        for i, j in ij:
            int1 = Jps[r, s, i, j]
            int2 = Kps[r, s, i, j]
            g += Ps[i, j] * (int1 - 0.5 * int2)
        G[r, s] = g
    return G


# Returns the Fock matrix
def Fp_matrix(fp, Z, Cos, Cop, Jpp, Kpp, Jps, Kps):
    return Hp_matrix(fp, Z) + Gp_matrix(Cos, Cop, Jpp, Kpp, Jps, Kps)


# slove secular equation, return the energy and improved coeffients
# the energy here is orbital energy for 1 electron
def secular_eqn(F, S):
    ei, C = eig(F, S)

    # sort eigvalue and eigvector from lower to higher
    idx = ei.argsort()[::1]
    ei = ei[idx]
    C = C[:, idx]

    # eigvector from scipy.linalg.eig is not normalized, which is a bug
    # this is to fix it
    Co = np.zeros((C.shape[0], C.shape[0]))
    inte = np.matmul(np.matmul(C.T, S), C)
    for i in range(C.shape[0]):
        for j in range(C.shape[0]):
            Co[j][i] = C[j][i] / np.sqrt(inte[i][i])
```

```
    return ei, Co


# return energy of atom
def get_E0(e1, e2, Ps, Pp, Hs, Hp):
    E0 = e1[0].real + e1[1].real + e2[0].real + 0.5 * (Ps * Hs).sum() + 0.5 *␣
 ↪(Pp * Hp).sum()
    return E0


# input
# zeta obtained from https://www.scm.com/zorabasis/periodic.dzae.html
zetas = [[13.45, 1], [9.60, 1], [4.05, 2], [2.55, 2], [1.15, 3], [0.65, 3]]
zetap = [[2.20, 2], [4.80, 2], [0.79, 3]]

# input nuclear charge (element number)
Z = 11

# build basis function
fs1 = STO(zetas[0][0], zetas[0][1])
fs2 = STO(zetas[1][0], zetas[1][1])
fs3 = STO(zetas[2][0], zetas[2][1])
fs4 = STO(zetas[3][0], zetas[3][1])
fs5 = STO(zetas[4][0], zetas[4][1])
fs6 = STO(zetas[5][0], zetas[5][1])
fs = [fs1, fs2, fs3, fs4, fs5,fs6]

fp1 = STO(zetap[0][0], zetap[0][1])
fp2 = STO(zetap[1][0], zetap[1][1])
fp3 = STO(zetap[2][0], zetap[2][1])
fp = [fp1, fp2, fp3]

# initialization
Jss = np.zeros((len(zetas), len(zetas), len(zetas), len(zetas)))
Kss = np.zeros((len(zetas), len(zetas), len(zetas), len(zetas)))
Jsp = np.zeros((len(zetas), len(zetas), len(zetap), len(zetap)))
Ksp = np.zeros((len(zetas), len(zetas), len(zetap), len(zetap)))
Jps = np.zeros((len(zetap), len(zetap), len(zetas), len(zetas)))
Kps = np.zeros((len(zetap), len(zetap), len(zetas), len(zetas)))
Jpp = np.zeros((len(zetas), len(zetas), len(zetap), len(zetap)))
Kpp = np.zeros((len(zetas), len(zetas), len(zetap), len(zetap)))
Hs = Hs_matrix(fs, Z)
Hp = Hp_matrix(fp, Z)
Ss = S_matrix(fs)
Sp = S_matrix(fp)
es, Cos = secular_eqn(Hs, Ss)
ep, Cop = secular_eqn(Hp, Sp)
```

```
Ps = Ps_matrix(Cos)
Pp = Pp_matrix(Cop)
scf_H = get_E0(es, ep, Ps, Pp, Hs, Hp)

################################################print information␣
 ↪below#################################################
print('-' * 30, "Initialization", '-' * 30)
print('-' * 25, "Ignore repulsion integral", '-' * 24)
display(Math(
    '\zeta_1 = {0} \quad \zeta_2 = {1} \quad \zeta_3 = {2} \quad \zeta_4 = {3}␣
 ↪\quad \zeta_5 = {4} \quad \zeta_6 = {5}'.format(
        format(zetas[0][0], '0.3f'), format(zetas[1][0], '0.3f'),␣
 ↪format(zetas[2][0], '0.3f'),
        format(zetas[3][0], '0.3f'), format(zetap[0][0], '0.3f'),␣
 ↪format(zetap[1][0], '0.3f'))))
display(Math('Orbitals:'))
display(Math(' \phi_1s = c_{11} \chi_1 + c_{21} \chi_2 + c_{31} \chi_3 + c_{41}␣
 ↪\chi_4'))
display(Math(' \phi_2s = c_{12} \chi_1 + c_{22} \chi_2 + c_{32} \chi_3 + c_{42}␣
 ↪\chi_4'))
display(Math(' \phi_2p = c_{52} \chi_5 + c_{62} \chi_6'))
display(Math('c11 = {0} \quad c21 = {1} \quad c31 = {2} \quad c41 = {3}'.
 ↪format(format(Cos[0][0], '0.3f'),
                                                                        ␣
 ↪format(Cos[1][0], '0.3f'),
                                                                        ␣
 ↪format(Cos[2][0], '0.3f'),
                                                                        ␣
 ↪format(Cos[3][0], '0.3f'))))
display(Math('c12 = {0} \quad c22 = {1} \quad c32 = {2} \quad c42 = {3}'.
 ↪format(format(Cos[0][1], '0.3f'),
                                                                        ␣
 ↪format(Cos[1][1], '0.3f'),
                                                                        ␣
 ↪format(Cos[2][1], '0.3f'),
                                                                        ␣
 ↪format(Cos[3][1], '0.3f'))))
display(Math('c52 = {0} \quad c62 = {1}'.format(format(Cop[0][0], '0.3f'),␣
 ↪format(Cop[1][0], '0.3f'))))

# plot density graph
colorlist = ['red', 'orange', 'yellow', 'green', 'blue', 'purple', 'black',␣
 ↪'red', 'orange', 'yellow', 'green', 'blue',
             'purple', 'black']
phi1 = Cos[0, 0] * fs1 + Cos[1, 0] * fs2 + Cos[2, 0] * fs3 + Cos[3, 0] * fs4
phi2 = Cos[0, 1] * fs1 + Cos[1, 1] * fs2 + Cos[2, 1] * fs3 + Cos[3, 1] * fs4
```

```
phi3 = Cop[0, 0] * fp1 + Cop[1, 0] * fp2
density_1 = phi1 * phi1 * r * r
density_2 = phi2 * phi2 * r * r
density_3 = phi3 * phi3 * r * r
p = plot((density_1, (r, 0, 5)), (density_2, (r, 0, 5)), (density_3, (r, 0,␣
↪5)), show=False, legend=True)
p[0].label = 'electron density $r^2 \phi_1^2$ '
p[1].label = 'electron density $r^2 \phi_2^2$ '
p[2].label = 'electron density $r^2 \phi_2^2$ '
p[0].line_color = colorlist[0]
p[1].line_color = 'blue'
p[2].line_color = 'green'
p.show()
# print energy result
display(Math(' \epsilon_1 \; for \; \phi_1  = {0} '.format(format(es[0].real,␣
↪'0.3f'))))
display(Math(' \epsilon_2 \; for \; \phi_2 = {0} '.format(format(es[1].real, '0.
↪3f'))))
display(Math(' \epsilon_3 \; for \; \phi_3 = {0} '.format(format(ep[0].real, '0.
↪3f'))))
display(Math(' Hartree \ Fork \; atom \; energy = {0} \ hartree = {1} \ eV'.
↪format(format(scf_H, '0.5f'),
                                                                          ␣
↪    format(scf_H * 27.211, '0.5f'))))

##################################################print information␣
↪above#################################################

for i in range(10):
    print('-' * 30, "Iteration", i + 1, '-' * 30)
    if (i == 0):
        print('-' * 7, "Iteration 1 needs more time to caculate Repulsion␣
↪Integral", '-' * 6)
        start = time.time()
        Jss = J_matrix(zetas, zetas, 0)
        Kss = K_matrix(zetas, zetas, 0)
        Jsp = J_matrix(zetas, zetap, 0)
        Ksp = 1 / 3 * K_matrix(zetas, zetap, 1)
        Jps = J_matrix(zetap, zetas, 0)
        Kps = 1 / 3 * K_matrix(zetap, zetas, 1)
        Jpp = J_matrix(zetap, zetap, 0) + 2 / 25 * J_matrix(zetap, zetap, 2)  #␣
↪1/25 for minimum
        Kpp = 1 / 5 * K_matrix(zetap, zetap, 2)  # 6/25 for minimum

    else:
        start = time.time()
```

```
    Fs = Fs_matrix(fs, Z, Cos, Cop, Jss, Kss, Jsp, Ksp)
    Fp = Fp_matrix(fp, Z, Cos, Cop, Jpp, Kpp, Jps, Kps)
    S = S_matrix(fs)
    es, Cos = secular_eqn(Fs, Ss)
    ep, Cop = secular_eqn(Fp, Sp)
    Ps = Ps_matrix(Cos)
    Pp = Pp_matrix(Cop)
    scf_H = get_E0(es, ep, Ps, Pp, Hs, Hp)
    ################################################print information␣
↪below#################################################
    # print information
    display(Math(
        '\zeta_1 = {0} \quad \zeta_2 = {1} \quad \zeta_3 = {2} \quad \zeta_4 =␣
↪{3} \quad \zeta_5 = {4} \quad \zeta_6 = {5}'.format(
            format(zetas[0][0], '0.3f'), format(zetas[1][0], '0.3f'),␣
↪format(zetas[2][0], '0.3f'),
            format(zetas[3][0], '0.3f'), format(zetap[0][0], '0.3f'),␣
↪format(zetap[1][0], '0.3f'))))
    display(Math('Orbitals:'))
    display(Math(' \phi_1s = c_{11} \chi_1 + c_{21} \chi_2 + c_{31} \chi_3 +␣
↪c_{41} \chi_4'))
    display(Math(' \phi_2s = c_{12} \chi_1 + c_{22} \chi_2 + c_{32} \chi_3 +␣
↪c_{42} \chi_4'))
    display(Math(' \phi_2p = c_{52} \chi_5 + c_{62} \chi_6'))
    display(Math('c11 = {0} \quad c21 = {1} \quad c31 = {2} \quad c41 = {3}'.
↪format(format(Cos[0][0], '0.3f'),
                                                                          ␣
↪    format(Cos[1][0], '0.3f'),
                                                                          ␣
↪    format(Cos[2][0], '0.3f'),
                                                                          ␣
↪    format(Cos[3][0], '0.3f'))))
    display(Math('c12 = {0} \quad c22 = {1} \quad c32 = {2} \quad c42 = {3}'.
↪format(format(Cos[0][1], '0.3f'),
                                                                          ␣
↪    format(Cos[1][1], '0.3f'),
                                                                          ␣
↪    format(Cos[2][1], '0.3f'),
                                                                          ␣
↪    format(Cos[3][1], '0.3f'))))
    display(Math('c52 = {0} \quad c62 = {1}'.format(format(Cop[0][0], '0.3f'),␣
↪format(Cop[1][0], '0.3f'))))
    # plot density graph
    phi1 = Cos[0, 0] * fs1 + Cos[1, 0] * fs2 + Cos[2, 0] * fs3 + Cos[3, 0] * fs4
    phi2 = Cos[0, 1] * fs1 + Cos[1, 1] * fs2 + Cos[2, 1] * fs3 + Cos[3, 1] * fs4
    phi3 = Cop[0, 0] * fp1 + Cop[1, 0] * fp2
```

```
    density_1 = phi1 * phi1 * r * r
    density_2 = phi2 * phi2 * r * r
    density_3 = phi3 * phi3 * r * r
    p = plot((density_1, (r, 0, 5)), (density_2, (r, 0, 5)), (density_3, (r, 0,␣
↪5)), show=False)
    p[0].label = 'electron density $r^2 \phi_1^2$ '
    p[1].label = 'electron density $r^2 \phi_2^2$ '
    p[2].label = 'electron density $r^2 \phi_2^2$ '
    p[0].line_color = colorlist[0]
    p[1].line_color = 'blue'
    p[2].line_color = 'green'
    p.show()
    # print energy result
    display(Math(' \epsilon_1 \; for \; \phi_1  = {0} '.format(format(es[0].
↪real, '0.3f'))))
    display(Math(' \epsilon_2 \; for \; \phi_2 = {0} '.format(format(es[1].
↪real, '0.3f'))))
    display(Math(' \epsilon_3 \; for \; \phi_3 = {0} '.format(format(ep[0].
↪real, '0.3f'))))
    display(Math(' Hartree \ Fork \; atom \; energy = {0} \ hartree = {1} \ eV'.
↪format(format(scf_H),
                                                                          ␣
↪        format(scf_H * 27.211, '0.5f'))))
    stop = time.time()
    print('Time used:', format(stop - start), 's')
    ##############################################print information␣
↪above#####################################################
```

```
------------------------------ Initialization ------------------------------
------------------------- Ignore repulsion integral -------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

Orbitals :

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

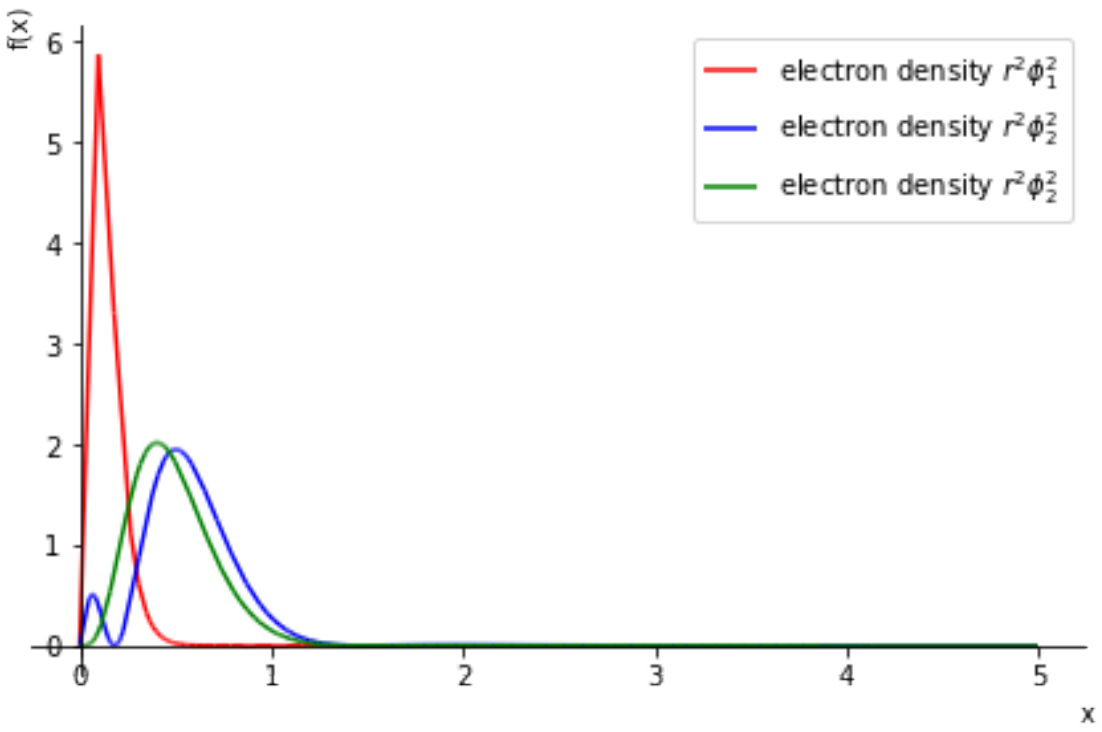
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.336 \quad c21 = 0.685 \quad c31 = -0.050 \quad c41 = 0.031$

$c12 = 0.004 \quad c22 = 0.419 \quad c32 = -1.544 \quad c42 = 0.530$

$c52 = 0.152 \quad c62 = -1.099$



$\epsilon_1\ for\ \phi_1 = -60.485$

$\epsilon_2\ for\ \phi_2 = -15.015$

$\epsilon_3\ for\ \phi_3 = -14.975$

$Hartree\ Fork\ atom\ energy = -180.95028\ hartree = -4923.83814\ eV$

```
------------------------------ Iteration 1 ------------------------------
------- Iteration 1 needs more time to caculate Repulsion Integral ------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

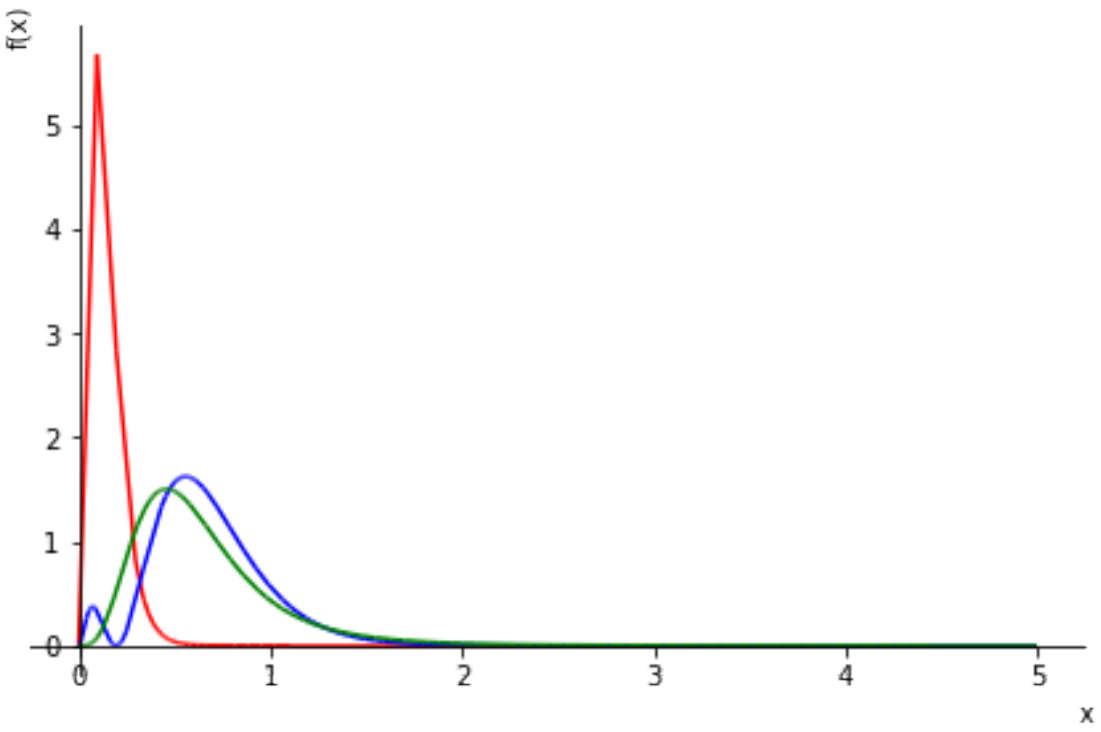
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.262 \quad c21 = 0.748 \quad c31 = -0.009 \quad c41 = 0.007$

$c12 = -0.070 \quad c22 = 0.445 \quad c32 = -1.028 \quad c42 = -0.076$

$c52 = -0.324 \quad c62 = -0.752$



$\epsilon_1\ for\ \phi_1 = -44.781$

$\epsilon_2\ for\ \phi_2 = -6.631$

$\epsilon_3\ for\ \phi_3 = -5.888$

$Hartree\ Fork\ atom\ energy = -146.27693644116115\ hartree = -3980.34172\ eV$

```
Time used: 1512.4210834503174 s
------------------------------ Iteration 2 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

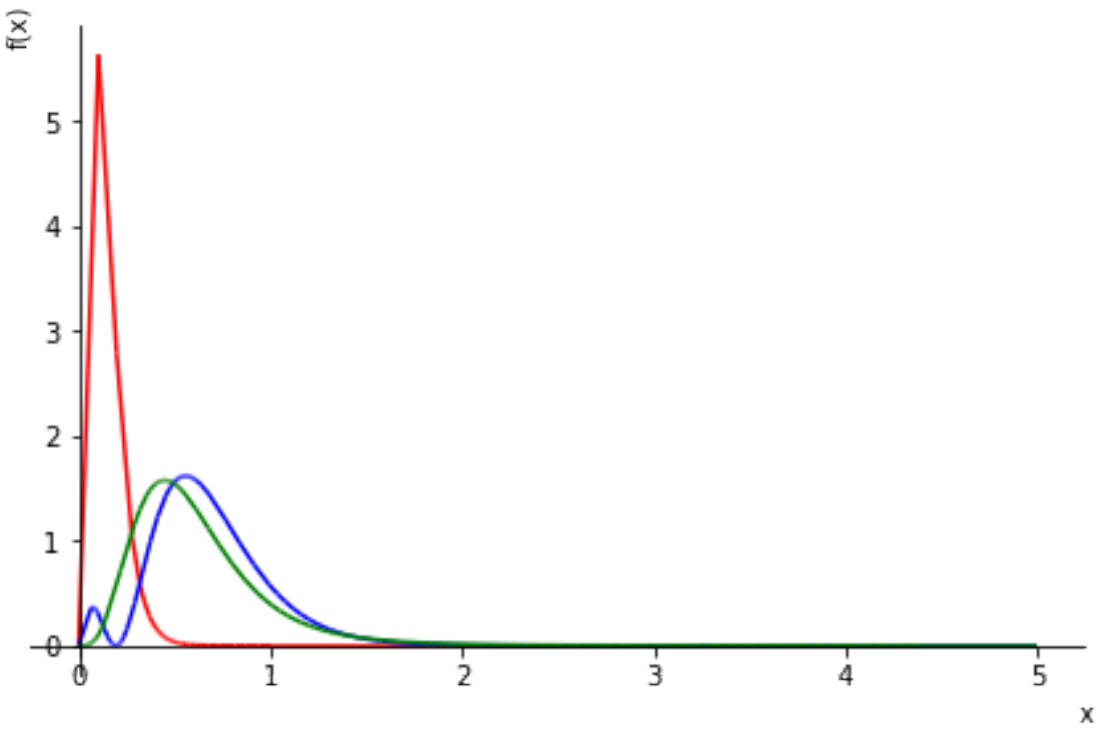
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.437 \quad c32 = -1.020 \quad c42 = -0.083$

$c52 = -0.266 \quad c62 = -0.800$



$\epsilon_1 \; for \; \phi_1 = -46.281$

$\epsilon_2 \; for \; \phi_2 = -7.085$

$\epsilon_3 \; for \; \phi_3 = -6.294$

$Hartree \; Fork \; atom \; energy = -148.83847715268712 \; hartree = -4050.04380 \; eV$

```
Time used: 1.627342939376831 s
------------------------------ Iteration 3 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

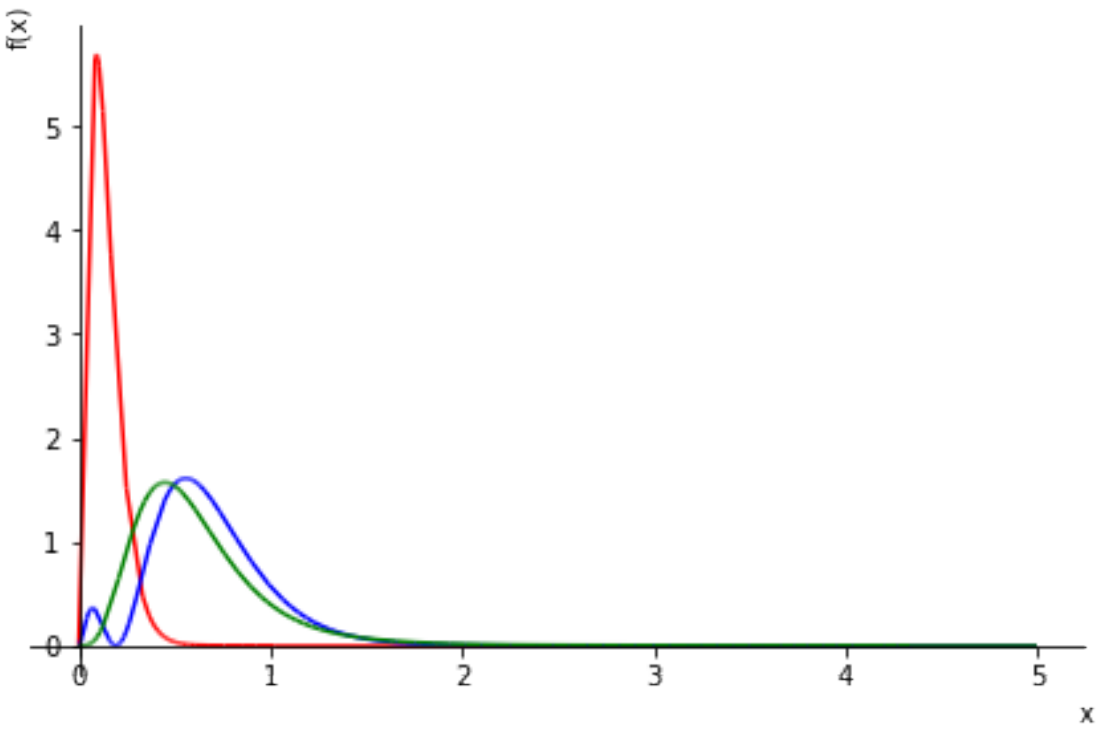
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.270 \quad c62 = -0.797$



$\epsilon_1\ for\ \phi_1 = -46.158$

$\epsilon_2\ for\ \phi_2 = -7.040$

$\epsilon_3\ for\ \phi_3 = -6.269$

$Hartree\ Fork\ atom\ energy = -148.61858720204114\ hartree = -4044.06038\ eV$

```
Time used: 1.5930736064910889 s
------------------------------ Iteration 4 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

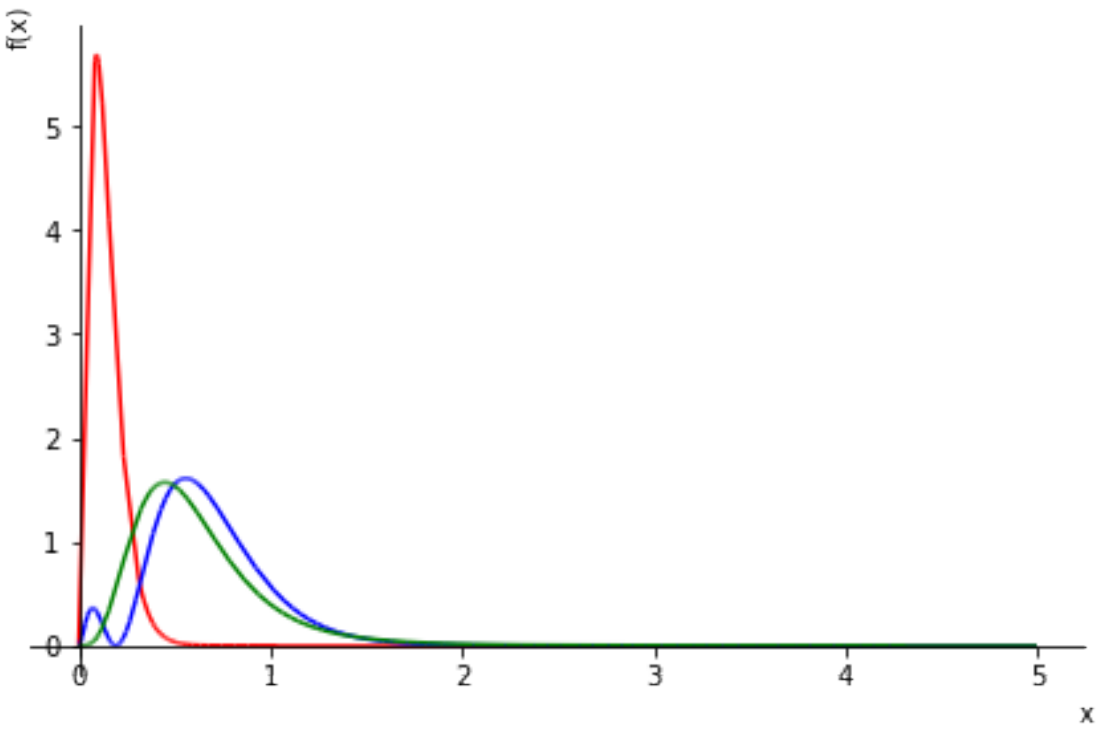
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.174$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64683554869532\ hartree = -4044.82904\ eV$

```
Time used: 1.5777535438537598 s
------------------------------ Iteration 5 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

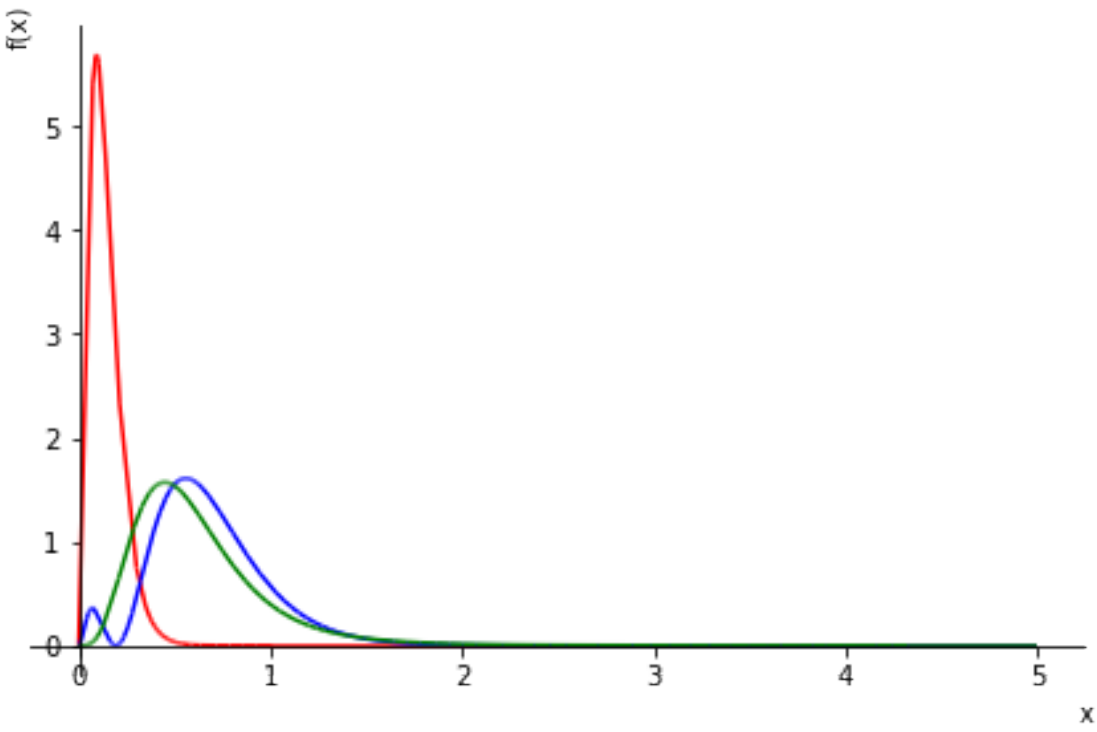
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64458451328858\ hartree = -4044.76779\ eV$

```
Time used: 1.5724573135375977 s
------------------------------ Iteration 6 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

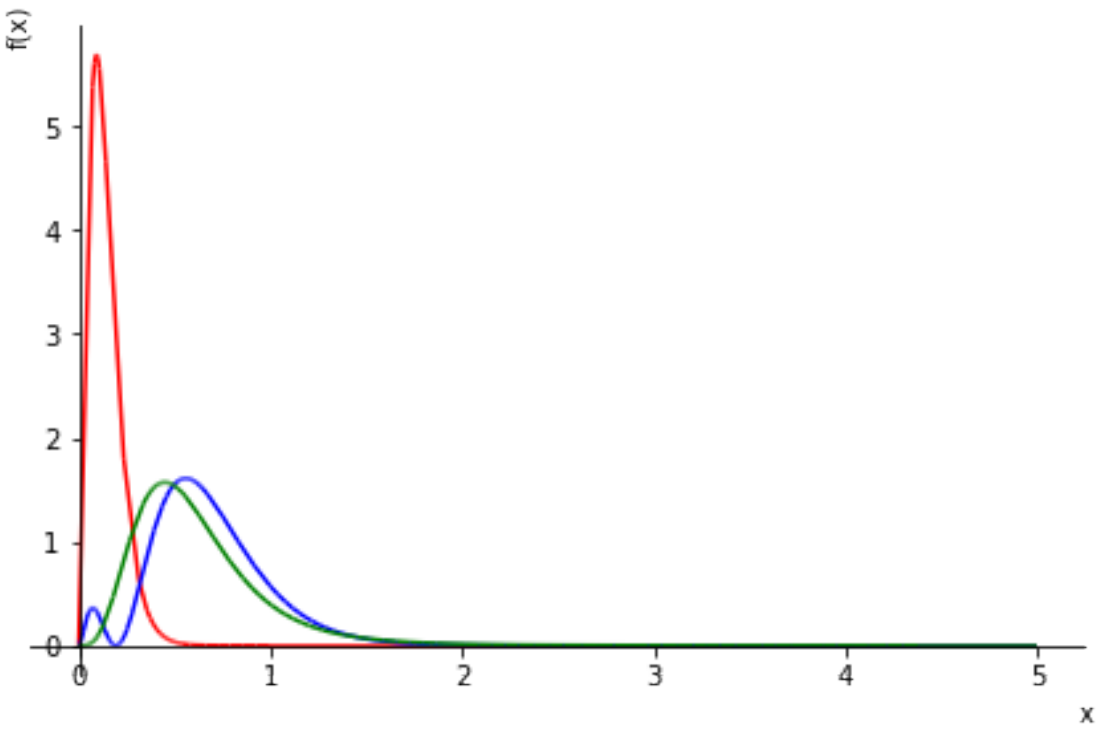
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64490163008378\ hartree = -4044.77642\ eV$

```
Time used: 1.5703306198120117 s
------------------------------ Iteration 7 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals:$

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

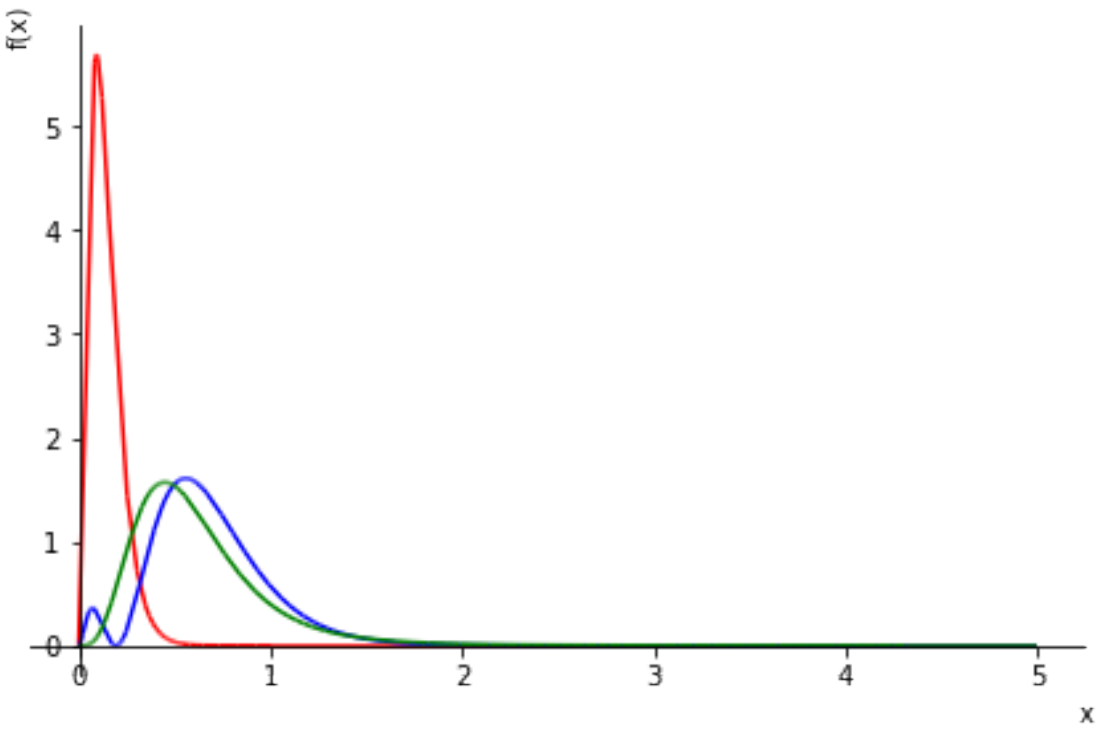
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64487909736965\ hartree = -4044.77581\ eV$

```
Time used: 1.5621416568756104 s
------------------------------ Iteration 8 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

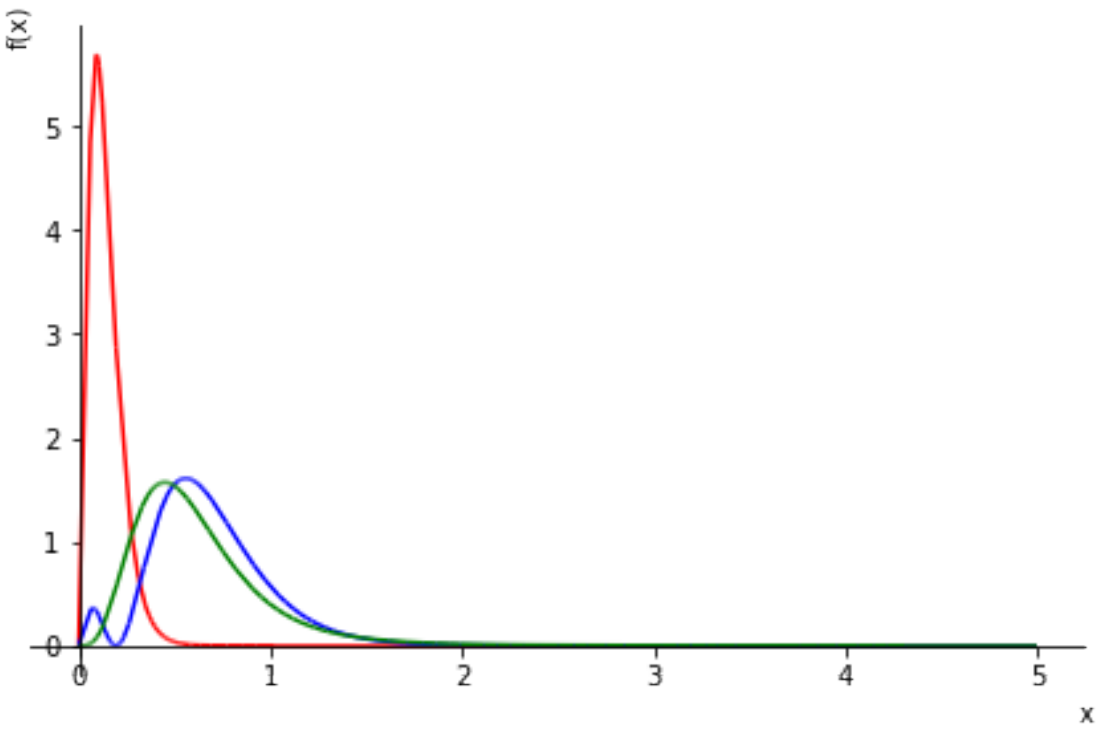
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64488271311066\ hartree = -4044.77590\ eV$

```
Time used: 1.5724208354949951 s
------------------------------ Iteration 9 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

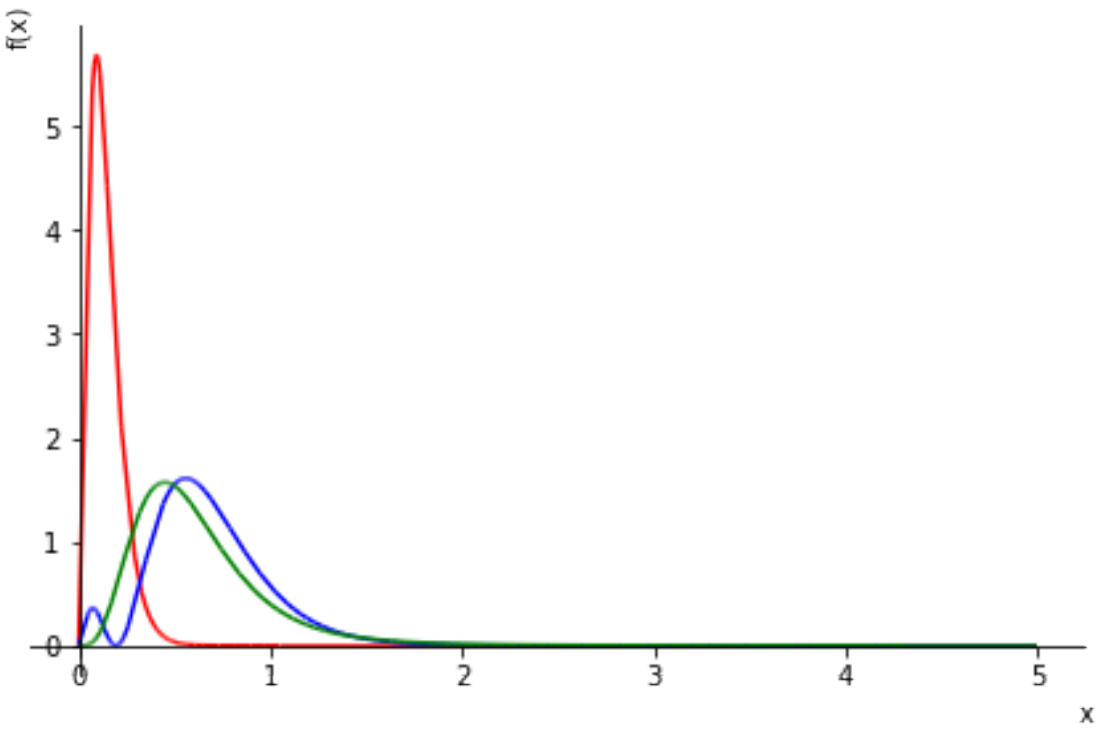
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.64488249634312\ hartree = -4044.77590\ eV$

```
Time used: 1.5679597854614258 s
------------------------------ Iteration 10 ------------------------------
```

$\zeta_1 = 13.450 \quad \zeta_2 = 9.600 \quad \zeta_3 = 4.050 \quad \zeta_4 = 2.550 \quad \zeta_5 = 2.200 \quad \zeta_6 = 4.800$

$Orbitals$:

$\phi_1 s = c_{11}\chi_1 + c_{21}\chi_2 + c_{31}\chi_3 + c_{41}\chi_4$

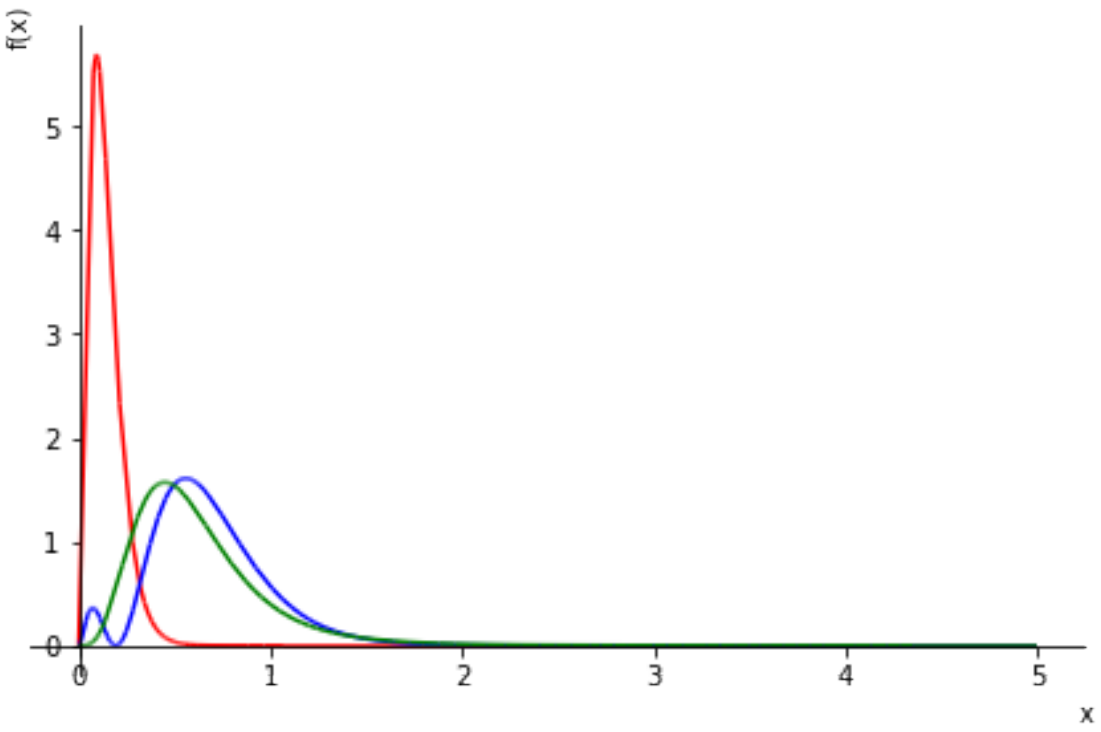
$\phi_2 s = c_{12}\chi_1 + c_{22}\chi_2 + c_{32}\chi_3 + c_{42}\chi_4$

$\phi_2 p = c_{52}\chi_5 + c_{62}\chi_6$

$c11 = 0.265 \quad c21 = 0.747 \quad c31 = -0.015 \quad c41 = 0.010$

$c12 = -0.066 \quad c22 = 0.436 \quad c32 = -1.012 \quad c42 = -0.091$

$c52 = -0.269 \quad c62 = -0.798$



$\epsilon_1\ for\ \phi_1 = -46.173$

$\epsilon_2\ for\ \phi_2 = -7.044$

$\epsilon_3\ for\ \phi_3 = -6.274$

$Hartree\ Fork\ atom\ energy = -148.6448825383679\ hartree = -4044.77590\ eV$

```
Time used: 1.5698485374450684 s
```

[ ]:

[ ]: