05_Atom_in_LDA_complete

May 22, 2023

1 Solution of the atom within LDA

In this excercise we will solve the multielectron atom in LDA approximation.

We will test it on He and oxygen by computing the total energy and charge density.

We will plot charge density and compute the total energy, which will be compared to the reference data at NIST database: https://www.nist.gov/pml/atomic-reference-data-electronic-structure-calculations/atomic-reference-data-electronic-7

We want to solve the Schroedinger equation for an atom with nucleous charge Z. We will approximate electron-electron interaction with an effective potential, which is computed by so-called "local density approximation" (LDA). In this theory, the classical (Hartree) potential is treated exactly, while the rest of the interaction is "approximated" by so called *local exchange-correlation functional*. We will not go into details of this functional, we will just use it here.

The Schroedinger equation we are solving is

$$(-\frac{\hbar^2}{2m}\nabla^2 - \frac{Ze^2}{4\pi\varepsilon_0 r} + V_H(r) + V_{xc}(r))\psi(\vec{r}) = E\psi(\vec{r}) \tag{1}$$

The first two terms are appearing in Hydrogen problem, and we already coded them. The Hartree is the electrostatic potential, and the exchange-correlation potential has an approximate form, which depends only the charge density of the system. We will use the module `excor.py`, where the function $V_{xc}(\rho)$ is tabulated. We will use it as $V_{xc}(r) == V_{xc}(\rho(r))$, corresponding to the local density approximation.

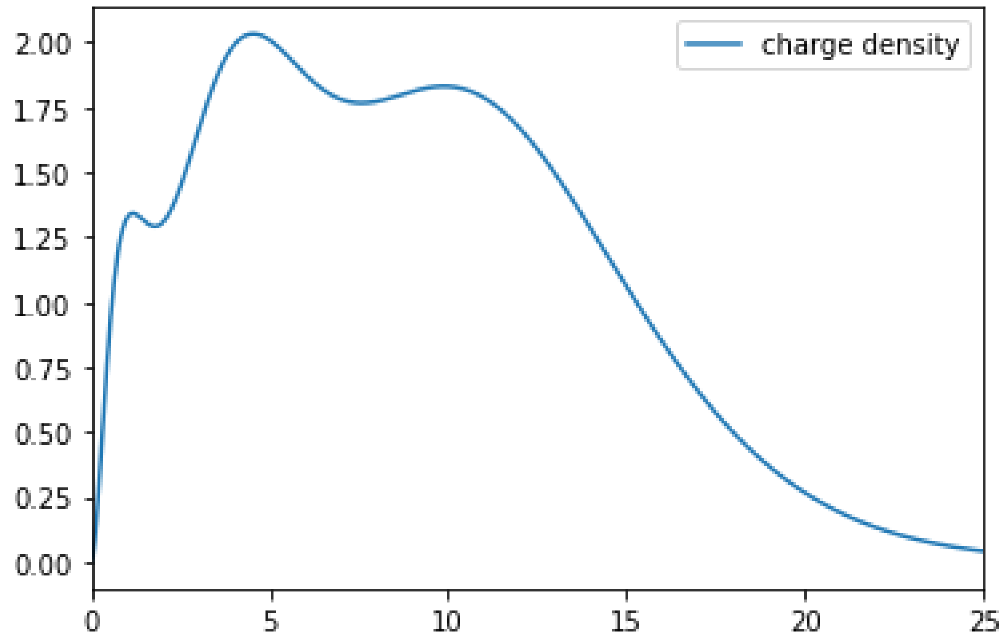
First we take the code from the Hydrogen project and repeat.

[1]:
```python
from scipy import *
from scipy import integrate
from scipy import optimize
from numpy import *
from pylab import *
from numba import jit
```

[2]:
```python
@jit(nopython=True)
def Numerov(f, x0, dx, dh):
    """ Given precumputed function f(x) solved the differential equation
        x''(t) = f(t) x(t)
        input: x0 = x(t=0), and dx = dx/dt(t=0)
    """
    x = zeros(len(f))
    x[0] = x0
    x[1] = x0 + dh*dx
    h2 = dh**2
    h12 = h2/12.
    w0 = x0*(1-h12*f[0])
    w1 = x[1]*(1-h12*f[1])
    xi = x[1]
    fi = f[1]
    for i in range(2,len(f)):
        w2 = 2*w1-w0 + h2*fi*xi
        fi = f[i]
        xi = w2/(1-h12*fi)
        x[i]=xi
        (w0,w1) = (w1,w2)
    return x

@jit(nopython=True)
def fShrod(En,l,R):
    return l*(l+1.)/R**2 - 2./R - En

def ComputeSchrod(En, R, l):
    f = fShrod(En,l,R[::-1])
    ur = Numerov(f, 0.0, -1e-7, -R[1]+R[0])[::-1]
    norm = integrate.simps(ur**2, x=R)
    return ur/sqrt(abs(norm))

def Shoot(En, R, l):
    ur = ComputeSchrod(En, R, l)
    ur = ur/R**l  # expecting urn \propto R
    f0,f1 = ur[0],ur[1]
    f_at_0 = f0 + (f1-f0)*(0-R[0])/(R[1]-R[0]) # extrapolation to zero
    return f_at_0

def FindBoundStates(R, l, nmax, Esearch):
    n=0
    Ebnd=[]
    u0 = Shoot(Esearch[0],R,l)
    for i in range(1,len(Esearch)):
        u1 = Shoot(Esearch[i],R,l)
        if u0*u1 < 0:
            #print 'Sign change at', Esearch[i-1], Esearch[i]
            Ebound = optimize.
 ↪brentq(Shoot,Esearch[i-1],Esearch[i],xtol=1e-15,args=(R,l))
            Ebnd.append( (l,Ebound) )
            if len(Ebnd)>nmax: break
            n += 1
            print('Found bound state at E=%14.9f' % Ebound)
        u0 = u1
    return Ebnd

def cmpKey(x):
    return x[1]*1000 + x[0]   # energy has large wait, but degenerate energy␣
 ↪states are sorted by l

def ChargeDensity(Bnd,R,Z):
    rho = zeros(len(R))
    N=0.
    for (l,En) in Bnd:
        ur = ComputeSchrod(En, R, l)
        dN = 2*(2*l+1)
        if N+dN <= Z:
            ferm = 1.
        else:
            ferm = (Z-N)/float(dN)
        drho = ur**2 * ferm * dN/(4*pi*R**2)
        rho += drho
        N += dN
        print('adding state', (l,En), 'with fermi=', ferm)
        if  N>=Z: break
    return rho
```

[3]:
```
%matplotlib inline

Esearch = -1.2/arange(1,20,0.2)**2
R = linspace(1e-8,100,2000)

Z=28
nmax = 5
Bnd=[]
for l in range(nmax-1):
    Bnd += FindBoundStates(R,l,nmax-l,Esearch)
Bnd = sorted(Bnd, key=cmpKey)

z = 28. # like Ni atom

rho = ChargeDensity(Bnd,R,Z)

plot(R, rho*(4*pi*R**2), label='charge density')
xlim([0,25])
legend(loc='best')
show()
```

```
Found bound state at E=  -0.999922109
Found bound state at E=  -0.249990190
Found bound state at E=  -0.111108201
Found bound state at E=  -0.062498772
Found bound state at E=  -0.039999314
Found bound state at E=  -0.250000016
Found bound state at E=  -0.111111117
Found bound state at E=  -0.062500003
Found bound state at E=  -0.039999959
Found bound state at E=  -0.111111111
Found bound state at E=  -0.062500000
Found bound state at E=  -0.039999977
Found bound state at E=  -0.062500000
Found bound state at E=  -0.039999992
adding state (0, -0.9999221089559623) with fermi= 1.0
adding state (0, -0.24999019020653063) with fermi= 1.0
adding state (1, -0.25000001561170354) with fermi= 1.0
adding state (0, -0.11110820082299863) with fermi= 1.0
adding state (1, -0.11111111678092336) with fermi= 1.0
adding state (2, -0.11111111114690274) with fermi= 1.0
```



1.1 Hartree term

The Hartree term is treated exactly in this approximation.

It describes the electrostatic interaction of one electron with the cloud of all electrons (including the electron itself). Mathematically, this term is

$$\frac{1}{2}\int d\vec{r}d\vec{r}'\psi^\dagger(\vec{r})\psi^\dagger(\vec{r}')v_c(\vec{r}-\vec{r}')\psi(\vec{r}')\psi(\vec{r}) \rightarrow \tag{2}$$
$$\int d\vec{r}\psi^\dagger(\vec{r})\psi(\vec{r})\int d\vec{r}'\langle\psi^\dagger(\vec{r}')\psi(\vec{r}')\rangle v_c(\vec{r}-\vec{r}') \equiv \int d\vec{r}\psi^\dagger(\vec{r})V_H(\vec{r})\psi(\vec{r})$$

with

$$V_H(\vec{r}) = 2\int d\vec{r}'\frac{\rho(\vec{r}')}{|\vec{r}-\vec{r}'|} \tag{3}$$

where 2 is due to Rydberg units sinc $v_c = 2/r$.

For any atom, the electron density is spherically symetric and hence V_H depends only on radial distance. (In solids, the hartree potential should be expanded in spheric harmonics to sufficiently high l, maybe $l = 6$).

1.1.1 Step 1: Using $\rho(r)$omputed in previous homework, compute the Hartree potential.

This is usually achieved by solving the Poisson equation. From clasical electrostatic we know

$$\nabla^2 V_H(\vec{r}) = -8\pi\rho(\vec{r}) \tag{4}$$

In Hartree approximation, we have

$$\frac{1}{r^2}\frac{d}{dr}(r^2\frac{dV_H}{dr}) = -8\pi\rho(r) \tag{5}$$

which simplifies to

$$U''(r) = -8\pi r\rho(r) \tag{6}$$

where $U(r) = V_H(r)r$.

This second order differential equation has the following boundary conditions $U(0) = 0$ and $U(\infty) = 2Z$.

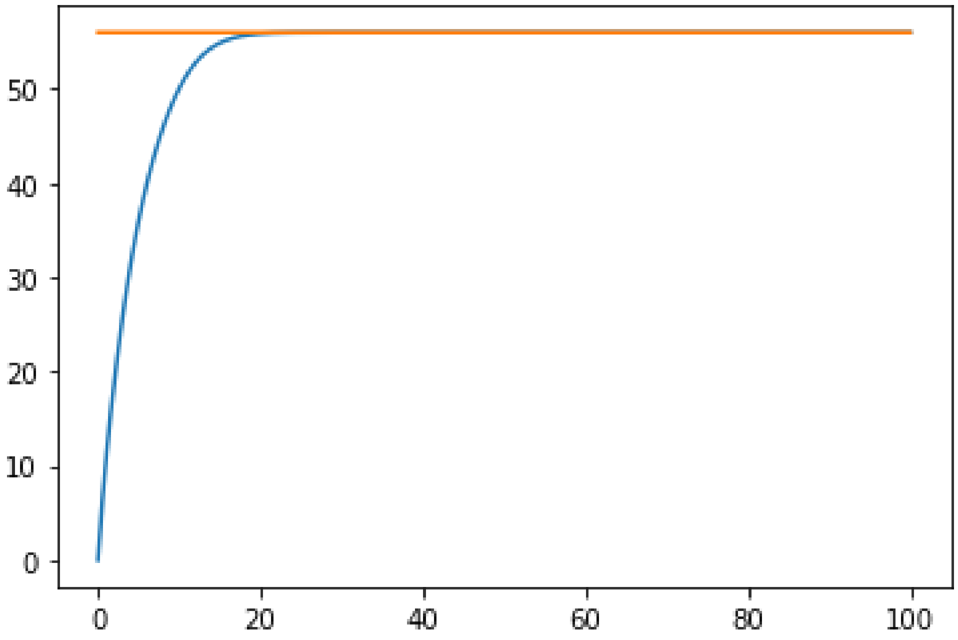
The two point boundary problem does not require shooting because we know solution to the homogenous differential equation $U''(r) = 0$. The Hartree potential can be obtained from any particular solution by

$$U(r) = U_p(r) + \alpha r \tag{7}$$

where $\alpha = \lim_{r\to\infty}(2Z - U_p(r))/r$.

[4]:
```
def FuncForHartree(y,r,rhoSpline):
    """ y = [U,U']
        dy/dr = [U', -8*pi*r*rho(r)]
    """
    return [y[1], -8*pi*r*rhoSpline(r)]
```

[5]:
```
from scipy import interpolate

rhoSpline = interpolate.UnivariateSpline(R, rho,s=0)

U0 = integrate.odeint(FuncForHartree, [0.0,0.5], R, args=(rhoSpline,))[:,0]
alpha = (2*Z - U0[-1])/R[-1]
U0 += alpha * R

plot(R, U0)
plot(R, ones(len(R))*2*Z)
show()
```



1.2 Numerov again

Poisson equation does not have the first order derivative, hence it can also be more efficiently solved by the Numerov algorithm.

We have Poisson equation, which has the form

$$x''(t) = u(t) \tag{8}$$

and the Numerov algorithm, as appropriate for the Poisson equation, is

$$x(h) + x(-h) = 2x(0) + h^2 u(0) + \frac{2}{4!} h^4 x^{(4)}(0) + O(h^6) \tag{9}$$

and the approximation for the forth order derivative is

$$x^{(4)} \sim \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \tag{10}$$

Inserting the fourth order derivative into the above recursive equation (forth equation in his chapter), we get

$$x_{i+1} - 2x_i + x_{i-1} = h^2 u_i + \frac{h^2}{12}(u_{i+1} - 2u_i + u_{i-1}) \tag{11}$$

If we switch to a new variable $w_i = x_i - \frac{h^2}{12} u_i$ we are left with the following equation

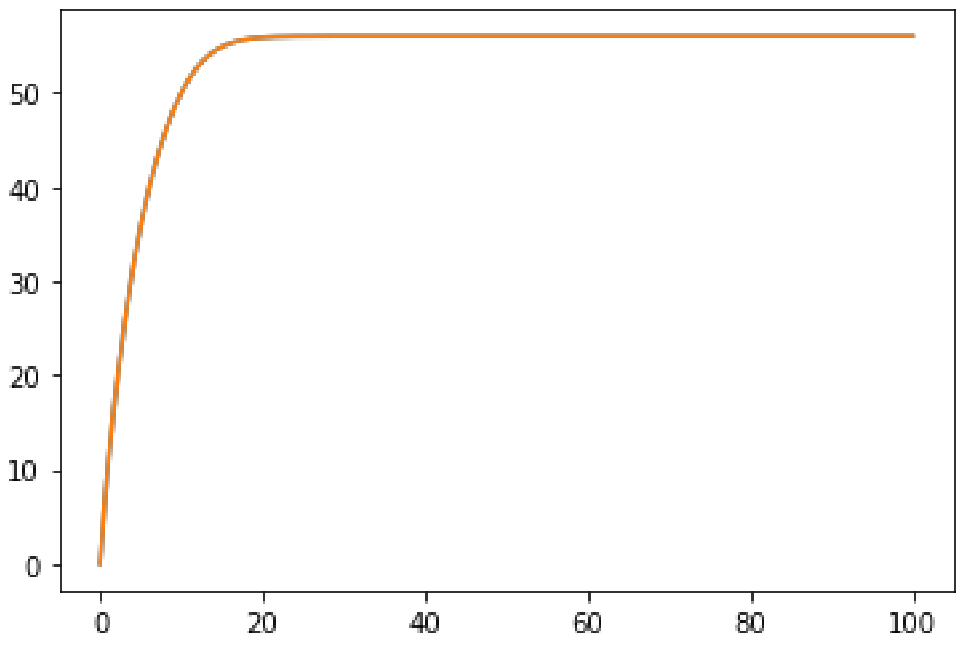
$$w_{i+1} - 2w_i + w_{i-1} = h^2 u_i + O(h^6) \tag{12}$$

The variable x needs to be recomputed at each step with $x_i = (w_i + \frac{h^2}{12} u_i)$.

[6]:
```
@jit(nopython=True)
def NumerovUP(U, x0, dx, dh):
    x = zeros(len(U))
    x[0] = x0
    x[1] = dh*dh + x0
    h2 = dh*dh
    h12 = h2/12
    w0 = x[0]-h12*U[0]
    w1 = x[1]-h12*U[1]
    Ui = U[1]
    for i in range(2,len(U)):
        w2 = 2*w1 - w0 + h2*Ui
        Ui = U[i]
        xi = w2 + h12*Ui
        x[i] = xi
        w0, w1 = w1, w2
    return x
```

[7]:
```
ux = -8*pi*R*rho
U2 = NumerovUP(ux, 0.0, 0.5, R[1]-R[0])
alpha2 = (2*Z-U2[-1])/R[-1]
U2 += alpha2 * R
```

[8]:
```
plot(R,U0)
plot(R,U2);
```

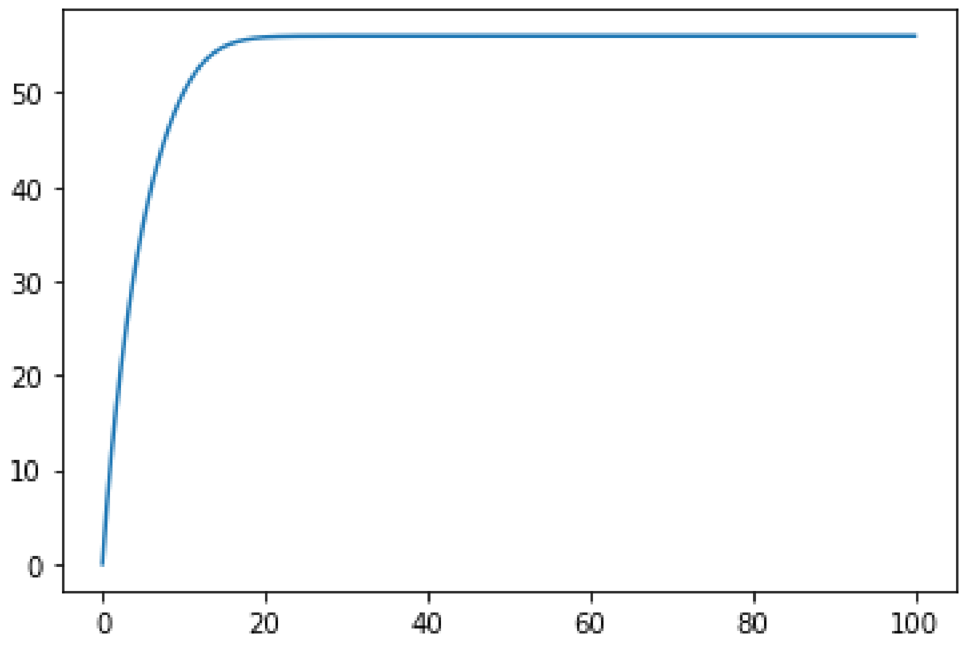


1.2.1 Step1 routine HartreeU

For generic density the following routine will work:

[9]:
```
def HartreeU(R, rho, Zatom):
    """Given input charge density it returns Hartree potential in the form␣
 ↪VH(r)*r
    """
    ux = -8*pi*R*rho
    U2 = NumerovUP(ux, 0.0, 0.5, R[1]-R[0])
    alpha2 = (2*Zatom-U2[-1])/R[-1]
    U2 += alpha2 * R
    return U2
```

[10]:
```
U2 = HartreeU(R,rho,Z)
plot(R,U2);
```



1.2.2 Step 2 : Compute the exchange correlation potential.

Note that $V_{xc}(r) = V_{xc}(\rho(r))$ is unquely determined by the electron charge density $\rho(r)$. If we know ρ, we can instantly compute V_{xc} by the module provided parametrized function.

Download the module `excor.py` from http://www.physics.rutgers.edu/~haule/509/src_prog/python/homework5/ and import it in your code.

Instantiate the ExchangeCorrelation object by

`exc = ExchangeCorrelation()`

and used it, for example, by

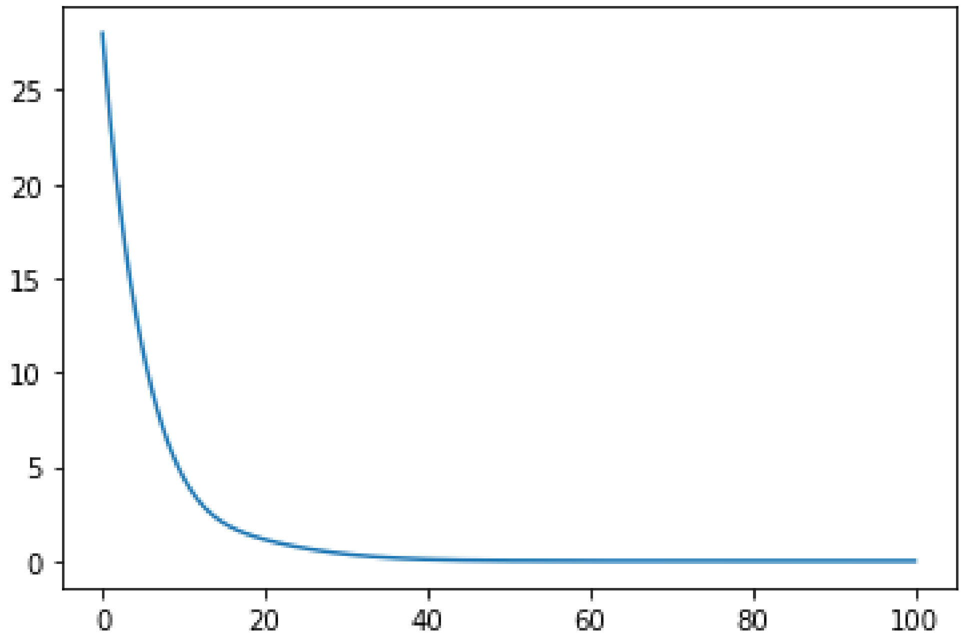
`exc.Vx(rs(rho[i]))+exc.Vc(rs(rho[i]))`

where $r_s = (4\pi\rho/3)^{-1/3}$.

Be careful: The energy unit in "excor.py" is Hartree and not Rydergs. Hence, you need to multiply energy or potential by 2.

[11]:
```
from excor import ExchangeCorrelation
exc = ExchangeCorrelation()

@jit(nopython=True)
def rs(rho):
    "1/rho = 4*pi*rs^3/3 => rs = (3/(4*pi*rho))**(1/3.)"
    if rho < 1e-100: return 1e100
    return pow(3/(4*pi*rho),1/3.)
```

[12]:
```
Vxc = [2*exc.Vc(rs(rh)) + 2*exc.Vx(rs(rh)) for rh in rho]

Uks = U2 - 2*Z + Vxc*R

plot(R, -Uks/2);
```



1.2.3 Step 3: Find bound states using Hartree and XC potential.

Add the Hartree potential and the exchange-correlation potential to the Schroedinger equation and find bound states of the atom.

The Schroedinger equation is

$$u''(r) = \left(\frac{l(l+1)}{r^2} - \frac{2Z}{r} + V_H(r) + V_{XC}(r) - \varepsilon\right) u(r). \tag{13}$$

or

$$u''(r) = \left(\frac{l(l+1)}{r^2} + \frac{U_H - 2Z + rV_{XC}}{r} - \varepsilon\right) u(r). \tag{14}$$

[13]:
```
@jit(nopython=True)
def fShrod(En,l,R, Uks):
    return (l*(l+1.)/R +Uks)/R - En

def ComputeSchrod(En, R, l,Uks):
    f = fShrod(En,l,R[::-1],Uks[::-1])
    ur = Numerov(f, 0.0, -1e-10, R[0]-R[1])[::-1]
    norm = integrate.simps(ur**2, x=R)
    return ur/sqrt(abs(norm))

def Shoot(En, R, l, Uks):
    ur = ComputeSchrod(En, R, l,Uks)
    ur *= 1/R**l  # expecting urn \propto R
    #f0,f1 = ur[0],ur[1]
    #f_at_0 = f0 + (f1-f0)*(0-R[0])/(R[1]-R[0]) # extrapolation to zero
    #return f_at_0
    poly = polyfit(R[:4], ur[:4], deg=3)
    return polyval(poly, 0.0)
```

[14]:
```
def FindBoundStates(R, l, nmax, Esearch, Uks):
    n=0
    Ebnd=[]
    u0 = Shoot(Esearch[0],R,l,Uks)
    for i in range(1,len(Esearch)):
        u1 = Shoot(Esearch[i],R,l,Uks)
        if u0*u1 < 0:
            Ebound = optimize.
 ↪brentq(Shoot,Esearch[i-1],Esearch[i],xtol=1e-15,args=(R,l,Uks))
            Ebnd.append( (l,Ebound) )
            if len(Ebnd)>nmax: break
            n += 1
            print('Found bound state at E=%14.9f' % Ebound)
        u0 = u1
    return Ebnd
```

[15]:
```
#R = linspace(1e-8,10,2**13+1)
#Zatom = 8
#E0=-1.2*Zatom**2
#Eshift=0.5 # sometimes energies can be positive!!!                              ␣
 ↪                                                                              ␣
 ↪
#Esearch = -logspace(-4,log10(-E0+Eshift),200)[::-1] + Eshift


Bnd=[]
for l in range(nmax-1):
    Bnd += FindBoundStates(R,l,nmax-l,Esearch,Uks)

Bnd = sorted(Bnd, key=cmpKey)
```

```
Found bound state at E=  -0.602399771
Found bound state at E=  -0.228625340
Found bound state at E=  -0.068176961
Found bound state at E=  -0.007230646
Found bound state at E=  -0.627631952
Found bound state at E=  -0.237253895
Found bound state at E=  -0.069506222
Found bound state at E=  -0.006696259
Found bound state at E=  -0.600054532
Found bound state at E=  -0.217328795
Found bound state at E=  -0.057702238
Found bound state at E=  -0.558268878
Found bound state at E=  -0.187465773
```

1.2.4 Step 4: Compute the new electron density

by filling the lowest Z eigenstatates.

[16]:
```
# This is modified from Hydrogen
def ChargeDensity(bst,R,Zatom,Uks):
    rho = zeros(len(R))
    N=0.
    Ebs=0.
    for (l,En) in bst:
        ur = ComputeSchrod(En, R, l, Uks)
        dN = 2*(2*l+1)
        if N+dN <= Zatom:
            ferm = 1.
        else:
            ferm = (Zatom-N)/float(dN)
        drho = ur**2 * ferm * dN/(4*pi*R**2)
        rho += drho
        N += dN
        Ebs += En * dN * ferm
        print('adding state', (l,En/2), 'H with fermi=', ferm)
        if  N>=Zatom: break
    return (rho,Ebs)
```

[17]:
```
rho_new, Ebs = ChargeDensity(Bnd,R,Z,Uks)
```

```
adding state (1, -0.31381597607349987) H with fermi= 1.0
adding state (0, -0.3011998855534427) H with fermi= 1.0
adding state (2, -0.3000272661852965) H with fermi= 1.0
adding state (3, -0.2791344391746787) H with fermi= 0.7142857142857143
```

1.2.5 Step 5: Admix the new and the old density

(50% of the old and 50% of the new should work) and use the resulting density to compute the new Hartree and exchange-correlation potential.

1.2.6 Iterate Steps 1 to Step 5 until self-consistency is achieved.

For oxygen, the total energy in this implementation is : -74.47303426133809 Hartree, while NIST shows -74.473077 Hartree. This is in excellent agreement. The small loss of accuracy is mainly due to the mesh used for R.

1.2.7 Evaluate the total energy

Once we see that the code converges, we can insert a new step betwen *Step 4* and *Step 5* to compute the total energy of the system. The total energy can be obtained by

$$\begin{aligned}
E^{LDA}_{total} &= \sum_{i\in occupied} \int d\vec{r}\psi^*_i(\vec{r})[-\nabla^2]\psi_i(\vec{r}) + \\
&+ \int d\vec{r}\rho(\vec{r})[V_{nucleous}(\vec{r}) + \epsilon_H(\vec{r}) + \epsilon_{XC}(\vec{r})] \\
&= \sum_{i\in occupied} \int d\vec{r}\psi^*_i(\vec{r})[-\nabla^2 + V_{nucleous} + V_H + V_{XC}]\psi_i(\vec{r}) \\
&+ \int d\vec{r}\rho(\vec{r})[\epsilon_H(\vec{r}) - V_H(\vec{r}) + \epsilon_{XC}(\vec{r}) - V_{XC}(\vec{r})] \\
&= \sum_{i\in occupied} \epsilon_i + \int d\vec{r}\rho(\vec{r})[\epsilon_H(\vec{r}) - V_H(\vec{r}) + \epsilon_{XC}(\vec{r}) - V_{XC}(\vec{r})] \\
&= \sum_{i\in occupied} \epsilon_i + \int d\vec{r}\rho(\vec{r})[-\epsilon_H(\vec{r}) + \epsilon_{XC}(\vec{r}) - V_{XC}(\vec{r})] \qquad (15)
\end{aligned}$$

Here we used

$$E_y[\rho] \equiv \int d\vec{r}\ \rho(\vec{r})\ \epsilon_y[\rho(\vec{r})] \qquad (16)$$

$$V_y[\rho] \equiv \frac{\delta E_y[\rho]}{\delta\rho(\vec{r})} \qquad (17)$$

where y is one of H, x or c.

Hence

$$\epsilon_H(\vec{r}) = \frac{1}{2}V_H(\vec{r}) = \frac{1}{2}\frac{U_H(\vec{r})}{r} \qquad (18)$$

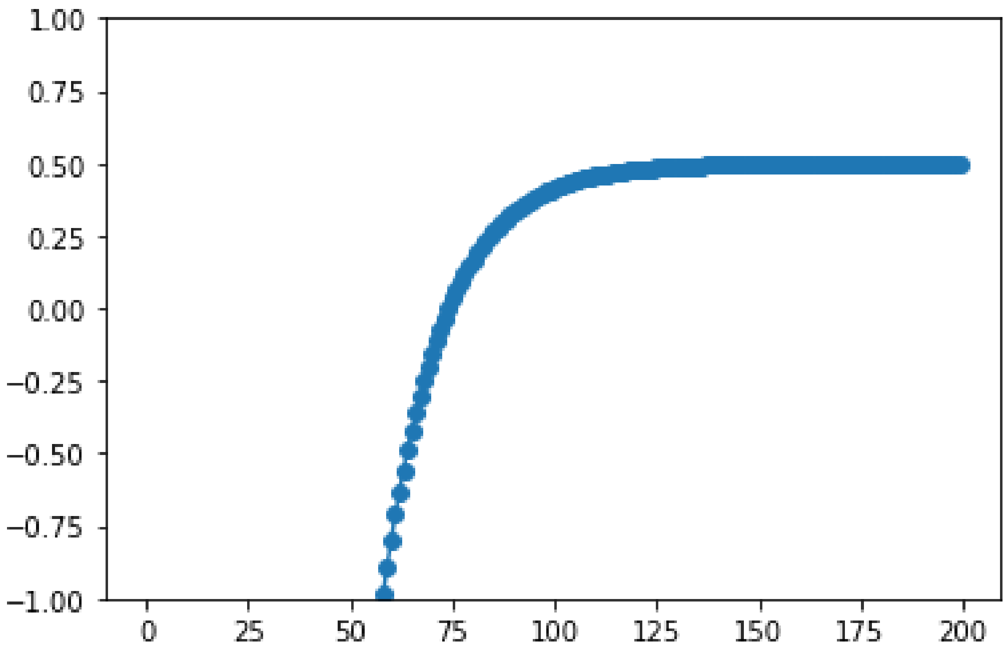
because

$$E_H = \int d\vec{r}d\vec{r}'\frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r}-\vec{r}'|} \qquad (19)$$

The exchange-correlation energy can be obtained by a call to the method of *ExchangeCorrelation* object.

[18]:
```
Zatom=8
E0=-1.2*Zatom**2
Eshift=0.5 # sometimes energies can be positive!!!                          ␣
  ↪                                                                        ␣
  ↪
Esearch = -logspace(-4,log10(-E0+Eshift),200)[::-1] + Eshift
plot(Esearch,'o-')
ylim([-1,1])
```

[18]:
```
(-1.0, 1.0)
```



[19]:
```
R = linspace(1e-8,20,2**14+1)
Zatom = 8
mixr = 0.5

E0=-1.2*Zatom**2
Eshift=0.5 # sometimes energies can be positive!!!                          ␣
  ↪                                                                        ␣
  ↪
Esearch = -logspace(-4,log10(-E0+Eshift),200)[::-1] + Eshift
nmax = 3

exc = ExchangeCorrelation()
Uks = -2*ones(len(R))
Eold = 0
```

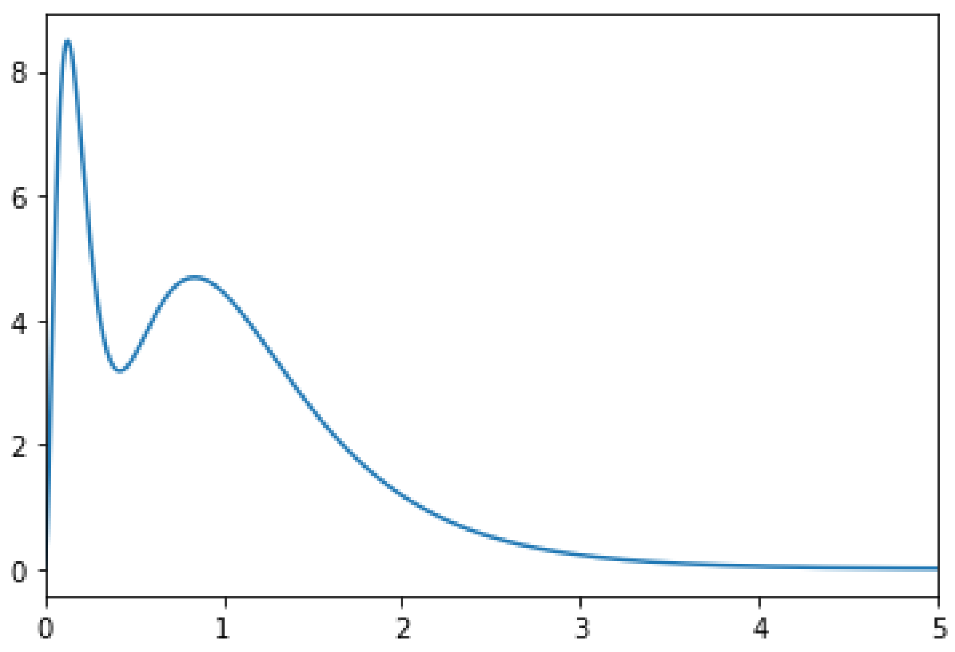
```
Etol = 1e-7

for itt in range(100):
    Bnd=[]
    for l in range(nmax-1):
        Bnd += FindBoundStates(R,l,nmax-l,Esearch,Uks)

    Bnd = sorted(Bnd, key=cmpKey)
    rho_new, Ebs = ChargeDensity(Bnd,R,Zatom,Uks)

    if itt==0:
        rho = rho_new
    else:
        rho = rho_new * mixr + (1-mixr)*rho_old
    rho_old = copy(rho)

    U2 = HartreeU(R, rho, Zatom)

    Vxc = [2*exc.Vc(rs(rh)) + 2*exc.Vx(rs(rh)) for rh in rho]

    Uks = U2 - 2*Zatom + Vxc*R

    # Total energy
    ExcVxc = [2*exc.EcVc(rs(rh)) + 2*exc.ExVx(rs(rh)) for rh in rho]
    pot = (ExcVxc*R - 0.5*U2)*R*rho*4*pi
    epot = integrate.simps(pot, x=R)
    Etot = epot + Ebs

    print('Total density has weight', integrate.simps(rho*(4*pi*R**2),x=R))
    #print('Total Energy=', Etot/2.)

    print('Itteration', itt, 'Etot[Ry]=', Etot, 'Etot[Hartre]=', Etot/2,␣
 ↪'Diff=', abs(Etot-Eold))


    if  itt>0 and abs(Etot-Eold)< Etol: break
    Eold = Etot

    #plot(R, U2, label='U-hartree')
    #plot(R, Vxc, label='Vxc')
    #plot(R, Uks, label='Uks')
    #show()

plot(R,rho*(4*pi*R**2))
xlim([0,5])
show()
```

```
Found bound state at E=  -1.000000000
Found bound state at E=  -0.249974229
Found bound state at E=  -0.099836095
Found bound state at E=  -0.249989213
Found bound state at E=  -0.103222840
adding state (0, -0.5000000000000026) H with fermi= 1.0
adding state (0, -0.12498711431297782) H with fermi= 1.0
adding state (1, -0.12499460664692974) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 0 Etot[Ry]= -16.72605139958573 Etot[Hartre]= -8.363025699792866 Diff=
16.72605139958573
Found bound state at E= -58.810318070
Found bound state at E= -11.072182717
Found bound state at E=  -2.936052943
Found bound state at E= -10.979642529
Found bound state at E=  -2.847650119
adding state (0, -29.40515903509979) H with fermi= 1.0
adding state (0, -5.536091358653621) H with fermi= 1.0
adding state (1, -5.489821264416604) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 1 Etot[Ry]= -224.73766490907386 Etot[Hartre]= -112.36883245453693
Diff= 208.01161350948811
Found bound state at E= -44.840026005
Found bound state at E=  -4.699936090
Found bound state at E=  -0.778811713
Found bound state at E=  -3.685736952
Found bound state at E=  -0.547793268
adding state (0, -22.420013002291636) H with fermi= 1.0
adding state (0, -2.3499680449095552) H with fermi= 1.0
adding state (1, -1.8428684758271663) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 2 Etot[Ry]= -173.90640224411695 Etot[Hartre]= -86.95320112205847
Diff= 50.83126266495691
Found bound state at E= -39.330036458
Found bound state at E=  -2.527354738
Found bound state at E=  -0.267077466
Found bound state at E=  -1.415419295
Found bound state at E=  -0.130982658
adding state (0, -19.66501822876814) H with fermi= 1.0
adding state (0, -1.2636773690170784) H with fermi= 1.0
adding state (1, -0.7077096477283258) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 3 Etot[Ry]= -154.6946219768114 Etot[Hartre]= -77.3473109884057 Diff=
19.211780267305556
Found bound state at E= -38.051551645
Found bound state at E=  -1.994459017
Found bound state at E=  -0.139164857
Found bound state at E=  -0.907351162
Found bound state at E=  -0.050735863
```

```
adding state (0, -19.025775822509868) H with fermi= 1.0
adding state (0, -0.9972295084059034) H with fermi= 1.0
adding state (1, -0.4536755811747082) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999999
Itteration 4 Etot[Ry]= -150.23794036808613 Etot[Hartre]= -75.11897018404306
Diff= 4.456681608725262
Found bound state at E= -37.776639816
Found bound state at E=  -1.866434817
Found bound state at E=  -0.088862327
Found bound state at E=  -0.790673332
Found bound state at E=  -0.019760826
adding state (0, -18.888319908206316) H with fermi= 1.0
adding state (0, -0.933217408322493) H with fermi= 1.0
adding state (1, -0.39533666614678353) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999999
Itteration 5 Etot[Ry]= -149.5761428352306 Etot[Hartre]= -74.7880714176153 Diff=
0.6617975328555303
Found bound state at E= -37.642706459
Found bound state at E=  -1.803163728
Found bound state at E=  -0.060565059
Found bound state at E=  -0.732655501
Found bound state at E=  -0.001671122
adding state (0, -18.821353229424847) H with fermi= 1.0
adding state (0, -0.9015818641334896) H with fermi= 1.0
adding state (1, -0.36632775058303746) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 6 Etot[Ry]= -149.25266799864875 Etot[Hartre]= -74.62633399932437
Diff= 0.32347483658185183
Found bound state at E= -37.577624490
Found bound state at E=  -1.772231724
Found bound state at E=  -0.042985374
Found bound state at E=  -0.704156394
Found bound state at E=   0.010225879
adding state (0, -18.788812245067994) H with fermi= 1.0
adding state (0, -0.8861158622126604) H with fermi= 1.0
adding state (1, -0.35207819682768654) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999999
Itteration 7 Etot[Ry]= -149.09499170269936 Etot[Hartre]= -74.54749585134968
Diff= 0.15767629594938626
Found bound state at E= -37.546125903
Found bound state at E=  -1.757174184
Found bound state at E=  -0.031381550
Found bound state at E=  -0.690226236
Found bound state at E=   0.018571441
adding state (0, -18.77306295128812) H with fermi= 1.0
adding state (0, -0.8785870918782738) H with fermi= 1.0
adding state (1, -0.3451131180636579) H with fermi= 0.6666666666666666
Total density has weight 8.0
```

```
Itteration 8 Etot[Ry]= -149.01867245500887 Etot[Hartre]= -74.50933622750443
Diff= 0.07631924769049192
Found bound state at E= -37.530859713
Found bound state at E=  -1.749821263
Found bound state at E=  -0.023471181
Found bound state at E=  -0.683397479
Found bound state at E=   0.024629933
adding state (0, -18.765429856379292) H with fermi= 1.0
adding state (0, -0.8749106316760735) H with fermi= 1.0
adding state (1, -0.34169873937952905) H with fermi= 0.6666666666666666
Total density has weight 8.000000000000002
Itteration 9 Etot[Ry]= -148.9815685469036 Etot[Hartre]= -74.4907842734518 Diff=
0.03710390810527997
Found bound state at E= -37.523458642
Found bound state at E=  -1.746219538
Found bound state at E=  -0.018008106
Found bound state at E=  -0.680040305
Found bound state at E=   0.029102768
adding state (0, -18.761729321074085) H with fermi= 1.0
adding state (0, -0.8731097689556807) H with fermi= 1.0
adding state (1, -0.34002015231036026) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 10 Etot[Ry]= -148.96348410766186 Etot[Hartre]= -74.48174205383093
Diff= 0.018084439241732753
Found bound state at E= -37.519869833
Found bound state at E=  -1.744449724
Found bound state at E=  -0.014233476
Found bound state at E=  -0.678385001
Found bound state at E=   0.032424891
adding state (0, -18.759934916528586) H with fermi= 1.0
adding state (0, -0.8722248620004903) H with fermi= 1.0
adding state (1, -0.3391925003879844) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 11 Etot[Ry]= -148.95464861531838 Etot[Hartre]= -74.47732430765919
Diff= 0.008835492343479245
Found bound state at E= -37.518129473
Found bound state at E=  -1.743577667
Found bound state at E=  -0.011645308
Found bound state at E=  -0.677566752
Found bound state at E=   0.034889152
adding state (0, -18.759064736460168) H with fermi= 1.0
adding state (0, -0.8717888334082475) H with fermi= 1.0
adding state (1, -0.33878337599178465) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 12 Etot[Ry]= -148.95032344415273 Etot[Hartre]= -74.47516172207636
Diff= 0.00432517116564668
Found bound state at E= -37.517285444
Found bound state at E=  -1.743146981
Found bound state at E=  -0.009893037
Found bound state at E=  -0.677161434
Found bound state at E=   0.036705087
adding state (0, -18.758642721916306) H with fermi= 1.0
adding state (0, -0.8715734905429611) H with fermi= 1.0
adding state (1, -0.3385807172037202) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999998
Itteration 13 Etot[Ry]= -148.94820274942776 Etot[Hartre]= -74.47410137471388
Diff= 0.002120694724965233
Found bound state at E= -37.516876060
Found bound state at E=  -1.742933892
Found bound state at E=  -0.008725080
Found bound state at E=  -0.676960340
Found bound state at E=   0.038029229
adding state (0, -18.75843803020282) H with fermi= 1.0
adding state (0, -0.8714669461000342) H with fermi= 1.0
adding state (1, -0.3384801701752102) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 14 Etot[Ry]= -148.9471615892203 Etot[Hartre]= -74.47358079461014
Diff= 0.0010411602074782422
Found bound state at E= -37.516677445
Found bound state at E=  -1.742828319
Found bound state at E=  -0.007959815
Found bound state at E=  -0.676860453
Found bound state at E=   0.038981762
adding state (0, -18.75833872232574) H with fermi= 1.0
adding state (0, -0.8714141595591143) H with fermi= 1.0
adding state (1, -0.33843022630985936) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 15 Etot[Ry]= -148.94664990988386 Etot[Hartre]= -74.47332495494193
Diff= 0.0005116793364265959
Found bound state at E= -37.516581039
Found bound state at E=  -1.742775958
Found bound state at E=  -0.007467222
Found bound state at E=  -0.676810792
Found bound state at E=   0.039656201
adding state (0, -18.75829051966772) H with fermi= 1.0
adding state (0, -0.8713879791252943) H with fermi= 1.0
adding state (1, -0.3384053958563243) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 16 Etot[Ry]= -148.94639820419718 Etot[Hartre]= -74.47319910209859
Diff= 0.0002517056866793155
Found bound state at E= -37.516534216
Found bound state at E=  -1.742749970
Found bound state at E=  -0.007155758
Found bound state at E=  -0.676786088
Found bound state at E=   0.040125405
adding state (0, -18.758267107853968) H with fermi= 1.0
adding state (0, -0.8713749851260902) H with fermi= 1.0
adding state (1, -0.33839304389321834) H with fermi= 0.6666666666666666
Total density has weight 8.000000000000002
Itteration 17 Etot[Ry]= -148.9462743114887 Etot[Hartre]= -74.47313715574435
Diff= 0.00012389270847279477
Found bound state at E= -37.516511451
Found bound state at E=  -1.742737061
Found bound state at E=  -0.006962274
Found bound state at E=  -0.676773790
Found bound state at E=   0.040445697
adding state (0, -18.758255725456358) H with fermi= 1.0
adding state (0, -0.8713685306335658) H with fermi= 1.0
adding state (1, -0.3383868950628522) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 18 Etot[Ry]= -148.94621324762886 Etot[Hartre]= -74.47310662381443
Diff= 6.106385984594453e-05
Found bound state at E= -37.516500375
Found bound state at E=  -1.742730650
Found bound state at E=  -0.006844169
Found bound state at E=  -0.676767670
Found bound state at E=   0.040659985
adding state (0, -18.758250187667187) H with fermi= 1.0
adding state (0, -0.8713653250013946) H with fermi= 1.0
adding state (1, -0.3383838350643333) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999999
Itteration 19 Etot[Ry]= -148.94618319477536 Etot[Hartre]= -74.47309159738768
Diff= 3.005285350354825e-05
Found bound state at E= -37.516494972
Found bound state at E=  -1.742727456
Found bound state at E=  -0.006773303
Found bound state at E=  -0.676764615
Found bound state at E=   0.040800370
adding state (0, -18.75824748613015) H with fermi= 1.0
adding state (0, -0.8713637281598529) H with fermi= 1.0
adding state (1, -0.338382307646775) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 20 Etot[Ry]= -148.94616833526595 Etot[Hartre]= -74.47308416763298
Diff= 1.4859509406051075e-05
Found bound state at E= -37.516492331
Found bound state at E=  -1.742725863
Found bound state at E=  -0.006731508
Found bound state at E=  -0.676763089
Found bound state at E=   0.040890331
adding state (0, -18.758246165674624) H with fermi= 1.0
adding state (0, -0.8713629316538264) H with fermi= 1.0
adding state (1, -0.3383815442527216) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 21 Etot[Ry]= -148.9461609560717 Etot[Hartre]= -74.47308047803585
```

```
Diff= 7.379194244094833e-06
Found bound state at E= -37.516491042
Found bound state at E=  -1.742725072
Found bound state at E=  -0.006707281
Found bound state at E=  -0.676762329
Found bound state at E=   0.040946663
adding state (0, -18.758245521167776) H with fermi= 1.0
adding state (0, -0.87136253599643) H with fermi= 1.0
adding state (1, -0.3383811643915223) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 22 Etot[Ry]= -148.94615732134224 Etot[Hartre]= -74.47307866067112
Diff= 3.6347294667393726e-06
Found bound state at E= -37.516490411
Found bound state at E=  -1.742724678
Found bound state at E=  -0.006693477
Found bound state at E=  -0.676761950
Found bound state at E=   0.040981100
adding state (0, -18.758245205713045) H with fermi= 1.0
adding state (0, -0.87136233894534) H with fermi= 1.0
adding state (1, -0.3383809748765246) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 23 Etot[Ry]= -148.94615552572526 Etot[Hartre]= -74.47307776286263
Diff= 1.7956169813260203e-06
Found bound state at E= -37.516490101
Found bound state at E=  -1.742724481
Found bound state at E=  -0.006685748
Found bound state at E=  -0.676761760
Found bound state at E=   0.041001632
adding state (0, -18.758245050724916) H with fermi= 1.0
adding state (0, -0.8713622404184704) H with fermi= 1.0
adding state (1, -0.3383808799500568) H with fermi= 0.6666666666666666
Total density has weight 7.999999999999998
Itteration 24 Etot[Ry]= -148.94615462720077 Etot[Hartre]= -74.47307731360038
Diff= 8.985244903669809e-07
Found bound state at E= -37.516489950
Found bound state at E=  -1.742724383
Found bound state at E=  -0.006681496
Found bound state at E=  -0.676761666
Found bound state at E=   0.041013558
adding state (0, -18.75824497512003) H with fermi= 1.0
adding state (0, -0.8713621917478453) H with fermi= 1.0
adding state (1, -0.33838083299698196) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 25 Etot[Ry]= -148.94615419225914 Etot[Hartre]= -74.47307709612957
Diff= 4.349416258264682e-07
Found bound state at E= -37.516489875
Found bound state at E=  -1.742724334
Found bound state at E=  -0.006679196
Found bound state at E=  -0.676761618
Found bound state at E=   0.041020304
adding state (0, -18.75824493757038) H with fermi= 1.0
adding state (0, -0.8713621671071636) H with fermi= 1.0
adding state (1, -0.33838080917667657) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 26 Etot[Ry]= -148.9461539711602 Etot[Hartre]= -74.4730769855801 Diff=
2.210989578088629e-07
Found bound state at E= -37.516489838
Found bound state at E=  -1.742724309
Found bound state at E=  -0.006677974
Found bound state at E=  -0.676761594
Found bound state at E=   0.041024020
adding state (0, -18.75824491902368) H with fermi= 1.0
adding state (0, -0.8713621547357939) H with fermi= 1.0
adding state (1, -0.33838079719815584) H with fermi= 0.6666666666666666
Total density has weight 8.0
Itteration 27 Etot[Ry]= -148.94615386155297 Etot[Hartre]= -74.47307693077649
Diff= 1.0960721397168527e-07
Found bound state at E= -37.516489819
Found bound state at E=  -1.742724297
Found bound state at E=  -0.006677334
Found bound state at E=  -0.676761582
Found bound state at E=   0.041026017
adding state (0, -18.758244909714534) H with fermi= 1.0
adding state (0, -0.8713621483890656) H with fermi= 1.0
adding state (1, -0.3383807910376456) H with fermi= 0.6666666666666666
Total density has weight 8.000000000000002
Itteration 28 Etot[Ry]= -148.94615380290628 Etot[Hartre]= -74.47307690145314
Diff= 5.864669105903886e-08
```


0
2
4
6
8
0
1
2
3
4
5

```
[ ]:
```

```
[ ]:
```