

Teoria do Funcional da Densidade

Soluções do Capítulo 5 (Thijssen, 2007) - Parte 1

7600065 - Mecânica Quântica Computacional

Henrique de Almeida Tórtura - henrique.tortura@usp.br

09/11/2020



IFSC UNIVERSIDADE
DE SÃO PAULO
Instituto de Física de São Carlos



Sumário

1. Revisão Teórica
2. Aplicações nos átomos de Hidrogênio e Hélio
3. Exercício 5.1 - Malha Não-Uniforme

Revisão Teórica

Unidades atômicas de Hartree

Constante	Valor aproximado no SI	Valor em u.a. de Hartree
\hbar	$1,055 \cdot 10^{-34} Js$	1 u.a. de ação
e	$1,602 \cdot 10^{-19} C$	1 u.a. de carga
a_0	$5,292 \cdot 10^{-11} m$	1 u.a. de comprimento
m_e	$9,109 \cdot 10^{-31} kg$	1 u.a. de massa

Energia: $1E_h = \frac{\hbar^2}{m_e a_0^2} \approx 27,211 eV$

- **Aproximação de Born-Oppenheimer:** resolver a equação de Schrödinger para elétrons se movendo no potencial gerado por núcleos.
- **Teoremas de Hohenberg-Kohn:** $\left\{ \begin{array}{l} 1^{\circ} \text{ Energia dada pela densidade eletrônica, } n \rightarrow H \rightarrow \Psi. \\ 2^{\circ} \text{ Densidade eletrônica do estado fundamental fornece a energia do estado fundamental, } E_0 = E[n_0]. \end{array} \right.$
- **Ansatz de Kohn-Sham:**
 $n_0(\mathbf{r})$ de um sist. interagente $\equiv n_0(\mathbf{r})$ de um sist. não-interagente.

Equações de Kohn-Sham

As (N) equações de Kohn-Sham,

$$\left[-\frac{1}{2}\nabla^2 + V_H(\mathbf{r}) + V_{XC}^\sigma(\mathbf{r}) + V_{ext}(\mathbf{r}) \right] \psi_k^\sigma(\mathbf{r}) = \epsilon_k^\sigma \psi_k^\sigma(\mathbf{r}), \quad (1)$$

sob a restrição (de autoconsistência)

$$n(\mathbf{r}) = \sum_{\sigma} \sum_{k=1}^{N_{\sigma}} |\psi_k^\sigma(\mathbf{r})|^2, \quad (2)$$

sendo a energia total

$$E[n] = \sum_{\sigma} \sum_{k=1}^{N_{\sigma}} \epsilon_k^\sigma - \frac{1}{2} \int V_H(\mathbf{r}) n(\mathbf{r}) d^3r - \int V_{XC}^\sigma(\mathbf{r}) n(\mathbf{r}) d^3r + E_{XC}^\sigma[n]. \quad (3)$$

Aproximação da Densidade Local (LDA)

Ansatz

$$E_{XC}^{LDA}[n] = \int n(\mathbf{r}) e_{XC}[n(\mathbf{r})] d^3r, \quad (4)$$

sendo e_{XC} a **energia (por partícula) de troca e correlação de um gás eletrônico homogêneo** com densidade $n(\mathbf{r})$ - útil para sistemas em que $n(\mathbf{r})$ não varia muito.

Normalmente, para $n(\mathbf{r}) = n^+(\mathbf{r}) + n^-(\mathbf{r})$,

$$E_{XC}^{LDA}[n] = -\frac{3^{4/3}}{4\pi^{1/3}} \int n^{4/3}(\mathbf{r}) d^3r + \int n(\mathbf{r}) \overbrace{e_c[n(\mathbf{r})]}^{\text{Depende da aproximação}} d^3r, \quad (5)$$

$$V_X^{LDA} = - \left[\frac{3n(\mathbf{r})}{\pi} \right]^{1/3}. \quad (6)$$

Aplicações nos átomos de Hidrogênio e Hélio

Hidrogênio

```
subroutine HydrogenAtom(r_range, Eigenvalue_range,&  
&KS_int_max, Eigenvalue_tol, u0_tol, Uniform_Numerov,&  
&h, j_max, delta)
```

Inputs (principais):

- **r_range**: limites radiais;
- **Eigenvalue_range**: limites de autovalores de KS;
- **KS_int_max**: limite do número de iterações para resolver KS;
- **Eigenvalue_tol**: tolerância na variação do autovalor;
- **u0_tol**: tolerância na variação da condição de contorno;
- **h**: discretização radial.

Hidrogênio - Resolvendo Kohn-Sham 1D

Vetores radial (uniforme) e potencial do núcleo

```
r(i) = r_range(1) + h*i  
Ext_Potential(i) = -1/r(i)
```

Chamada da subrotina “KohnSham1D” com inputs/outputs **u** e **Eigenvalue**

```
call KohnSham1D(r, u, Ext_Potential, Eigenvalue_Range,&  
&Eigenvalue, KS_int_max, Eigenvalue_tol, u0_tol,&  
&n, h, rp, delta, Uniform_Numerov)
```

Chutes iniciais $u(r) = r\psi_R(r) = (cte)re^{-r}$ no infinito

```
u(n) = r(n)*exp(-r(n))  
u(n-1) = r(n-1)*exp(-r(n-1))
```

Hidrogênio - Kohn-Sham 1D - Condição de Contorno

```
EigenValue = (Eigenvalue_Range(1)+Eigenvalue_Range(2))/2
u = Numerov(u, -2*(EigenValue-Potential), h, n)

if ((abs(u(1))>=u0_tol .or.&
&abs(EigenValue-Eigenvalue_Range(1))>=eigenvalue_tol)) then
    if (u(1)<0) then
        Eigenvalue_Range(2) = EigenValue
    else
        Eigenvalue_Range(1) = EigenValue
    end if
else
    exit
end if
```

Hidrogênio - Kohn-Sham 1D - Numerov

Numerov aplicado a equação $\frac{d^2 u(r)}{dr^2} = -2 [\epsilon - V_{ef}^\sigma(r)] u(r) = f(r)u(r)$

```
function Numerov(u, f, h, n)
...
    do i=2, n-1
        a = 2*(1 + 5*(h**2)*f(n-i+1)/12)
        b = (1 - (h**2)*f(n-i+2)/12)
        c = (1 - (h**2)*f(n-i)/12)
        u(n-i) = (a*u(n-i+1) - b*u(n-i+2)) / c
    end do
```

Finalmente, normalizamos u

```
u = u / sqrt(sum(u*u*h))
```

Hidrogênio - Resolvendo Poisson

Potencial auxiliar $U(r) = rV_H(r)$

```
call Poisson(r, u, Potential_U, n, h, rp, delta,&  
&Uniform_Numerov)
```

Chutes iniciais $U(0) = 0$ e $U(h) = h$ (como orientado no Thijssen, 2007)

```
Potential_U(1) = r(1)  
Potential_U(2) = r(2)
```

Hidrogênio - Poisson - Verlet

Verlet aplicado a $\frac{d^2U(r)}{dr^2} = -\frac{u^2(r)}{r} = F(r)$

```
Potential_U = Verlet(Potential_U , -u**2/r , h , n)
...
function Verlet(Potential_U , F , h , n)
...
    do i=3 , n
        Potential_U(i) = 2*Potential_U(i-1)&
            &- Potential_U(i-2) + (h**2)*F(i-1)
    end do
```

Adicionamos uma solução da homogênea (ar) para garantir $U(r_{max}) = 1$

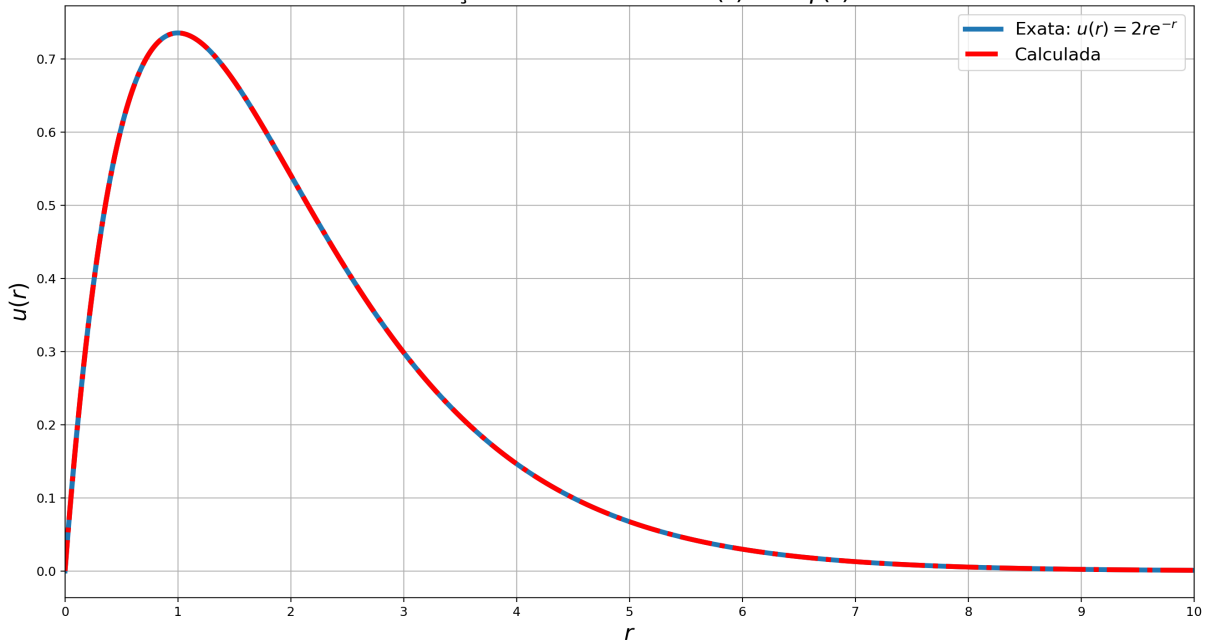
```
a = (1 - Potential_U(n)) / r(n)
Potential_U = Potential_U + a*r
```

Hidrogênio - Resultados

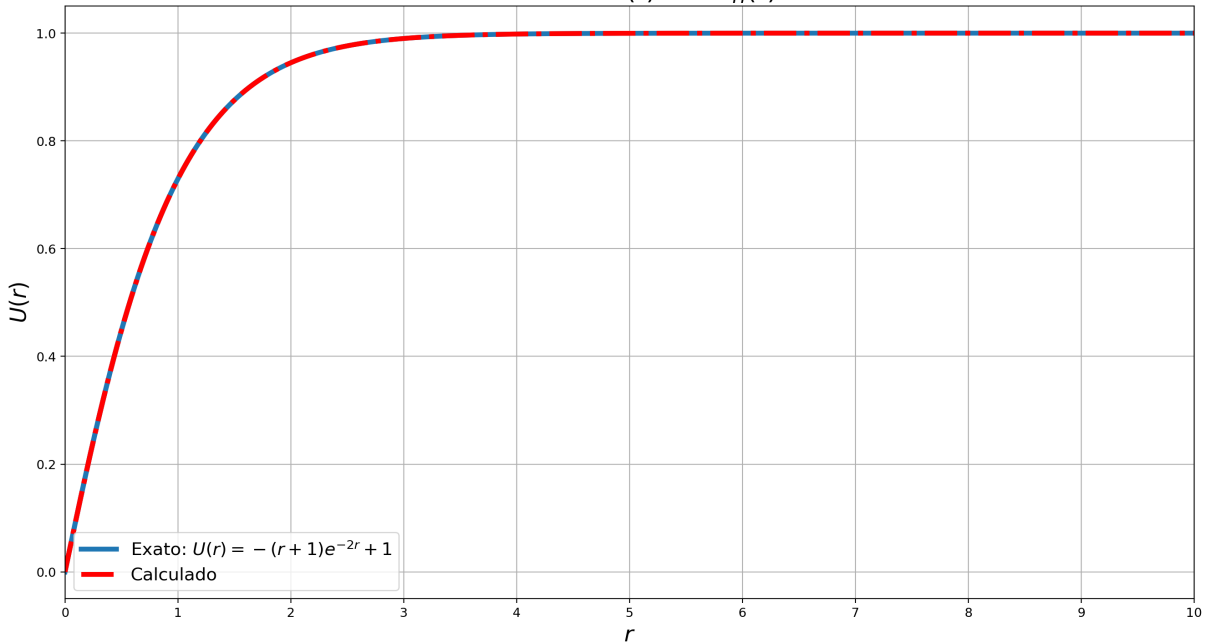
```
r_range = (/ 0., 50./), Eigenvalue_Range = (/ -5., 0./)  
u0_tol = 0.001
```

<i>eigenvalue_tol</i>	<i>h</i>	<i>u</i> (0)	Erro (abs) do autovalor	cpu_time (s)
10^{-3}	10^{-3}	$-2,64 \cdot 10^{-4}$	$2,56 \cdot 10^{-3}$	0,02
10^{-3}	10^{-5}	$-5,07 \cdot 10^{-5}$	$1,22 \cdot 10^{-4}$	1,23
10^{-5}	10^{-3}	$2,43 \cdot 10^{-6}$	$1,96 \cdot 10^{-3}$	0,02
10^{-5}	10^{-5}	$1,41 \cdot 10^{-6}$	$1,72 \cdot 10^{-5}$	1,78

Função de onda radial $u(r) = r \cdot \psi(r)$



Potencial $U(r) = r \cdot V_H(r)$



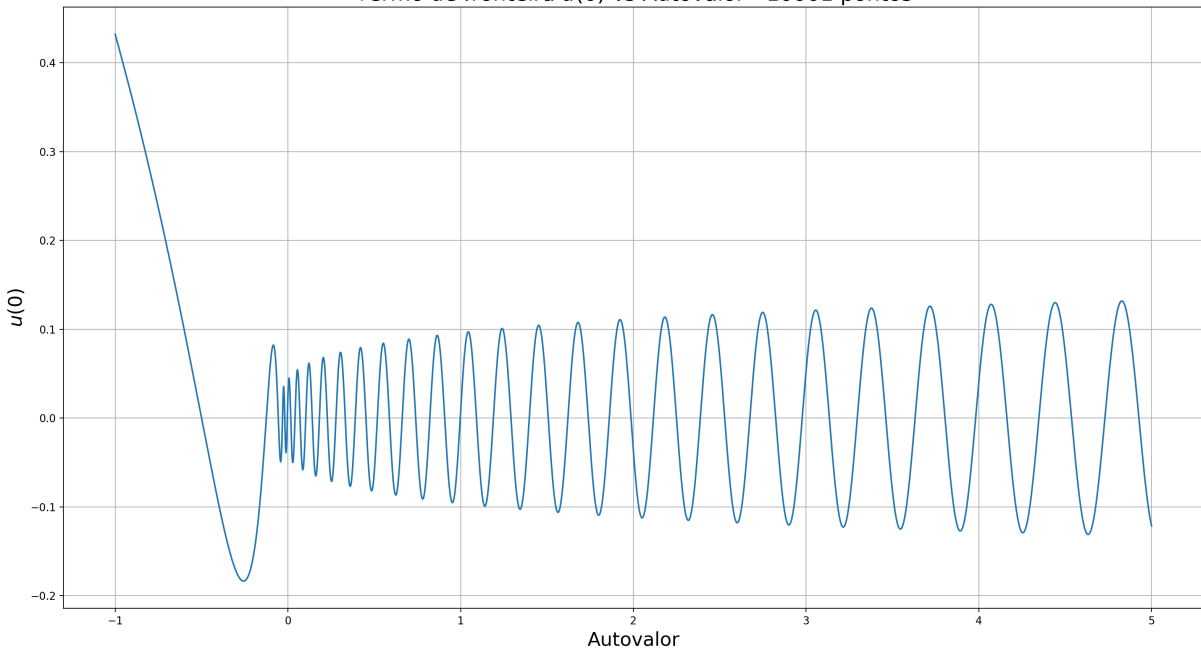
Hidrogênio - Varrendo Autovalores

Termo de fronteira $u(0)$ para $n_eigenvalues$ pontos de autovalores

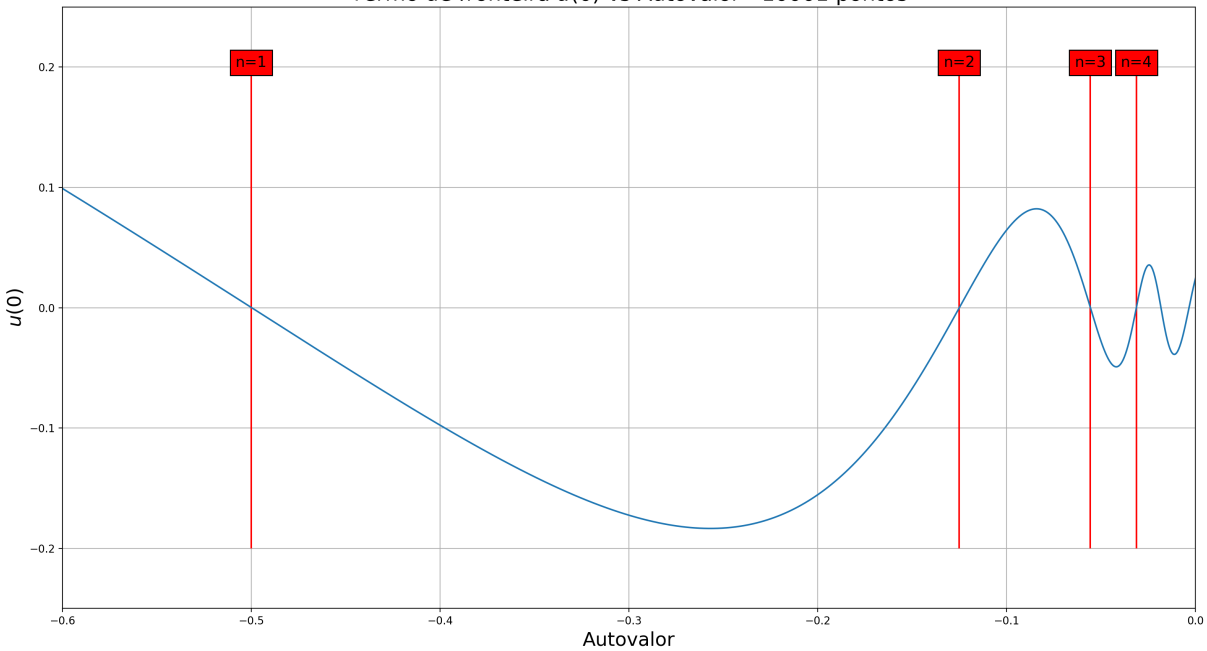
```
do i=1, n_eigenvalues
  Eigenvalues(i) = Eigenvalue_Range(1) +&
    &(i-1)*(Eigenvalue_Range(2)-Eigenvalue_Range(1))&
    &/(n_eigenvalues-1)
end do
call KohnShamSweep(r, u, Potential, h, Eigenvalues, u0s)
```

```
do i=1, n_eigenvalues
  u = Numerov(u, -2*(Eigenvalues(i)-Potential), h, n)
  u = u/sqrt(sum(u*u*h))
  u0s(i) = u(1)
end do
```

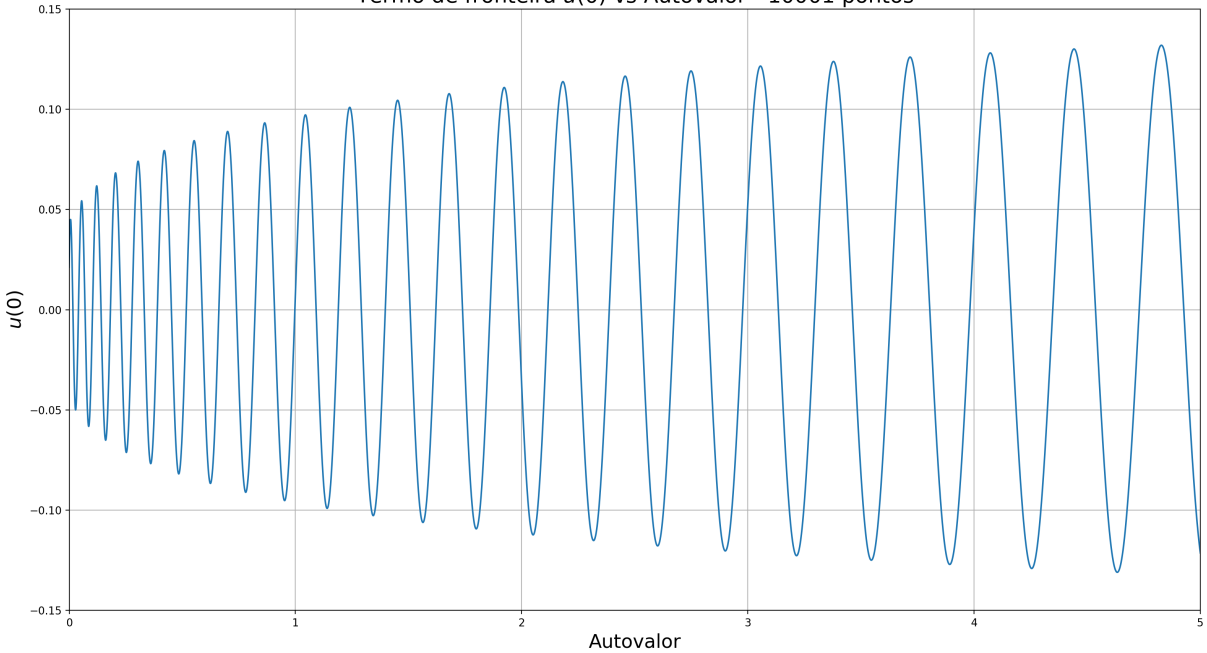
Termo de fronteira $u(0)$ vs Autovalor - 10001 pontos



Termo de fronteira $u(0)$ vs Autovalor - 10001 pontos



Termo de fronteira $u(0)$ vs Autovalor - 10001 pontos



```
subroutine HeliumAtom(r_range, Eigenvalue_range,&  
&SelfCons_int_max, KS_int_max, Eigenvalue_tol, u0_tol,&  
&SelfCons_tol, Uniform_Numerov, h, j_max, delta)
```

Inputs (principais e novos):

- **SelfCons_int_max**: limite do número de iterações de autoconsistência;
- **SelfCons_tol**: tolerância na variação do autovalor numa iteração de autoconsistência.

Potencial externo e chutes iniciais para o de Hartree e de troca

```
Ext_Potential(i) = -2/r(i)  
Hartree(i) = 0  
Exchange(i) = 0
```

Hélio - Laço de autoconsistência

Potencial efetivo, $V_{ef}^{\sigma}[n] = V_H[n] + V_{XC}^{\sigma}[n] + V_{ext}(\mathbf{r})$, na solução de Kohn-Sham

```
do i=1, SelfCons_int_max
  E_Range_aux = Eigenvalue_Range
  Eigenvalue_aux = Eigenvalue

  call KohnSham1D(r, u, Ext_Potential+Hartree+Exchange,&
    &E_Range_aux, Eigenvalue, KS_int_max, Eigenvalue_tol,&
    &u0_tol, n, h, rp, delta, Uniform_Numerov)
```

Hélio - Laço de autoconsistência

Se a tolerância não é satisfeita, atualizamos o potencial de Hartree e de troca

```
if (abs(Eigenvalue-Eigenvalue_aux)>=SelfCons_tol) then
  call Poisson(r, u, Potential_U, n, h, rp, delta,&
    &Uniform_Numerov)
  Hartree = 2 * Potential_U / r

  Exchange = -((3./2.)*(u/(pi * r))**2.)*(1.0/3.0)
else
  exit
end if
end do
```


Enfim, obtemos a energia

```
Energy = 2.*Eigenvalue - sum(Hartree*(u**2.)*h)&  
&- (1./2.)*sum(Exchange*(u**2.)*h)
```

Correção de auto-interação inicialmente sugerida pelo Thijssen

```
Hartree = Potential_U / r  
Exchange = 0
```

“For an integration step $h = 0.01$ (in the Verlet algorithm) you will find for the eigenvalue of the radial Schrödinger equation the value -0.923 a.u. and for the Hartree correction 1.0155 a.u., so that the total energy amounts to $E = -2.861$ a.u., in good agreement with the result obtained in the previous chapter.” (Thijssen, 2007, p. 113)

Hélio - Resultados - Sem potencial de troca

```
Eigenvalue_tol = 0.00001, u0_tol = 0.00001,&  
&SelfCons_tol = 0.0001
```

Esperado: autovalor de $-0,923$ u.a. e energia de $-2,861$ u.a.

h	Autovalor	Energia	cpu_time (s)
10^{-2}	-0.858	$-2,675$	0,05
10^{-4}	$-0,917$	$-2,859$	4,53
10^{-5}	$-0,918$ (não convergiu em 100 iterações)	$-2,861$	73,43 🌟

Hélio - Resultados - Com potencial de troca

```
Hartree = 2 * Potential_U / r  
Exchange = -((3./2.)*(u/(pi * r))**2.)*(1.0/3.0)
```

“Check If your program is correct, it should give the following values for the energies: $\epsilon = -0.52$ and $E = -2.72$ a.u.” (Thijssen, 2007, p. 114)

Usando $h = 10^{-4}$,

- Autovalor: **0,51652193069458008**
- Energia: **-2,7214406731542931**
- cpu_time: 3,49744606 s

Exercício 5.1 - Malha Não-Uniforme

5.1 - a)

Como $r(j) := r_j = r_p (e^{j\delta} - 1)$, $j = 0, 1, \dots, j_{max}$,

$$dr = r_p \delta e^{j\delta} dj \rightarrow \frac{dj}{dr} = \frac{e^{-j\delta}}{r_p \delta}, \quad (7)$$

a equação de Schrödinger radial fica

$$\begin{aligned} \frac{d^2}{dr^2} u(r) &= [V(r) - E] u(r) \\ \frac{e^{-j\delta}}{r_p \delta} \frac{d}{dj} \left(\frac{e^{-j\delta}}{r_p \delta} \frac{du}{dj} \right) &= [V(j) - E] u(j) \\ -\delta e^{-j\delta} \frac{du}{dj} + e^{-j\delta} \frac{d^2 u}{dj^2} &= r_p^2 \delta^2 e^{j\delta} [V(j) - E] u(j) \end{aligned} \quad (8)$$

$$\therefore \boxed{\frac{d^2 u}{dj^2} - \delta \frac{du}{dj} = r_p^2 \delta^2 e^{2j\delta} [V(j) - E] u(j)} \text{ Q.E.D.} \quad (9)$$

5.1 - b)

Usando a Equação 7, temos

$$\int_0^{r_{max}} f(r) dr = \boxed{r_p \delta \int_0^{j_{max}} e^{j\delta} f(j) dj}, \quad (10)$$

sendo

$$r_{max} = r_p (e^{j_{max}\delta} - 1) \rightarrow \boxed{j_{max} = \frac{1}{\delta} \ln \left(\frac{r_{max}}{r_p} + 1 \right)}. \quad (11)$$

▷ Resolvemos Kohn-Sham:

$$\frac{d^2 u(r)}{dr^2} = -2 [\epsilon - V_{ef}^\sigma(r)] u(r) \quad (12)$$

$$\boxed{\frac{d^2 u}{dj^2} - \delta \frac{du}{dj} = -2r_p^2 \delta^2 e^{2j\delta} [\epsilon - V_{ef}^\sigma(j)] u(j)}$$

$$\text{Cond. Inic.} \begin{cases} u(j_{max}) = r(j_{max}) e^{-r(j_{max})} \\ \left(\frac{du}{dr} \frac{dr}{dj} \right) \Big|_{j_{max}} = \frac{du}{dj}(j_{max}) = [1 - r(j_{max})] e^{-r(j_{max})} r_p \delta e^{j_{max}\delta} \end{cases} \quad (13)$$

5.1 - c) Runge-Kutta em Kohn-Sham

$$\begin{cases} \boxed{\frac{du}{dj} = z} = f(j, u, z) \\ \boxed{\frac{dz}{dj} = \underbrace{-2r_p^2 \delta^2}_c e^{2j\delta} \underbrace{[\epsilon - V_{ef}^\sigma(j)]}_{\tilde{V}} u(j) + \delta z} = g(j, u, z) \end{cases} \quad (14)$$

$$\begin{cases} K_0 = hf(j, u_j, z_j) \\ K_1 = hf\left(j + \frac{h}{2}, u_j + \frac{K_0}{2}, z_j + \frac{L_0}{2}\right) \\ K_2 = hf\left(j + \frac{h}{2}, u_j + \frac{K_1}{2}, z_j + \frac{L_1}{2}\right) \\ K_3 = hf(j + h, u_j + K_2, z_j + L_2) \end{cases} \quad \begin{cases} L_0 = hg(j, u_j, z_j) \\ L_1 = hg\left(j + \frac{h}{2}, u_j + \frac{K_0}{2}, z_j + \frac{L_0}{2}\right) \\ L_2 = hg\left(j + \frac{h}{2}, u_j + \frac{K_1}{2}, z_j + \frac{L_1}{2}\right) \\ L_3 = hg(j + h, u_j + K_2, z_j + L_2) \end{cases} \quad (15)$$

5.1 - c) Runge-Kutta em Kohn-Sham

$$h = -1$$

$$\left\{ \begin{array}{l} K_0 = -z(j) \\ K_1 = -\left[z(j) + \frac{L_0}{2}\right] \\ K_2 = -\left[z(j) + \frac{L_1}{2}\right] \\ K_3 = -\left[z(j) + \frac{L_2}{2}\right] \end{array} \right. \quad \left\{ \begin{array}{l} L_0 = -ce^{2j\delta}\tilde{V}(j)u(j) - \delta z(j) \\ L_1 = -ce^{2(j-1/2)\delta} \left[\frac{\tilde{V}(j)+\tilde{V}(j-1)}{2} \right] \left[u(j) + \frac{K_0}{2} \right] - \delta \left[z(j) + \frac{L_0}{2} \right] \\ L_2 = -ce^{2(j-1/2)\delta} \left[\frac{\tilde{V}(j)+\tilde{V}(j-1)}{2} \right] \left[u(j) + \frac{K_1}{2} \right] - \delta \left[z(j) + \frac{L_1}{2} \right] \\ L_3 = -ce^{2(j-1)\delta}\tilde{V}(j-1) \left[u(j) + K_2 \right] - \delta \left[z(j) + L_2 \right] \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{du}{dj} \Big|_{j-1} = \frac{du}{dj} \Big|_j + \frac{L_0+2L_1+2L_2+L_3}{6} \\ \boxed{u(j-1) = u(j) + \frac{K_0 + 2K_1 + 2K_2 + K_3}{6}} \end{array} \right. \quad (16)$$

5.1 - c) Runge-Kutta em Poisson

▷ Resolvemos Poisson (Potencial de Hartree)

$$\begin{aligned}\frac{d^2 U(r)}{dr^2} &= -\frac{u^2(r)}{r} \\ \frac{d^2 U}{dj^2} - \delta \frac{dU}{dj} &= -r_p^2 \delta^2 e^{2j\delta} \frac{u^2(j)}{r_p (e^{j\delta} - 1)} \\ \boxed{\frac{d^2 U}{dj^2} - \delta \frac{dU}{dj} &= -\frac{r_p \delta^2}{(e^{-j\delta} - e^{-2j\delta})} u^2(j)} \quad (17) \\ &\downarrow\end{aligned}$$

Resolvido também com Runge-Kutta

5.1 - c) Código

Kohn-Sham

```
du(n) = (1 - r(n))*exp(-r(n))*rp*delta*exp(n*delta)
...
else
    u = RungeKutta_KohnSham(u, du,&
        &-2*(EigenValue-Potential), rp**2*delta**2, delta, n)
```

Poisson

```
else
    dPotential_U(1) = (2*r(1)+1)*exp(-2*r(1))*rp*delta*&
    &exp(n*delta)
    Potential_U = RungeKutta_Poisson(Potential_U,&
    &dPotential_U, -u**2, rp*delta**2, delta, n)
```

5.1 - c) Resultados - Hidrogênio

```
r_range = (/ 0., 50./), Eigenvalue_Range = (/ -5., 0./)
eigenvalue_tol = 0.00001, u0_tol= 0.001
h = 0.0001
j_max = 500000, delta = 0.0001
```

Malha	$u(0)$	Erro (abs) do autovalor	cpu_time (s)
Uniforme	$4,122 \cdot 10^{-6}$	$2,079 \cdot 10^{-4}$	0,18
Não-uniforme	$3,371 \cdot 10^{-6}$	$4,005 \cdot 10^{-5}$	0,76

5.1 - c) Resultados - Hélio

```
r_range = (/ 0., 20./), Eigenvalue_Range = (/ -5., 0./)
Eigenvalue_tol = 0.0001, u0_tol = 0.0001,&
&SelfCons_tol = 0.0001
h = 0.0001
j_max = 200000, delta = 0.0001
```

Malha	$u(0)$	Erro (abs) do autovalor	Erro (abs) da energia	cpu_time (s)
Uniforme	$8,107 \cdot 10^{-5}$	$3,452 \cdot 10^{-3}$	$1,509 \cdot 10^{-3}$	0,97
Não-uniforme	$-2,955 \cdot 10^{-5}$	$1,355 \cdot 10^{-3}$	$1,026 \cdot 10^{-2}$ ($E = -2,730$)	4,03

5.1 - d)

Substituindo $u(j) = v(j)e^{\frac{j\delta}{2}}$ na Equação 9, temos

$$\frac{d^2}{dj^2} \left[v(j)e^{\frac{j\delta}{2}} \right] - \delta \frac{d}{dj} \left[v(j)e^{\frac{j\delta}{2}} \right] = r_p^2 \delta^2 e^{2j\delta} [V(j) - E] v(j)e^{\frac{j\delta}{2}}$$

$$\left[\frac{\delta^2}{4} e^{\frac{j\delta}{2}} v(j) + \cancel{\delta e^{\frac{j\delta}{2}} \frac{dv}{dj}} + e^{\frac{j\delta}{2}} \frac{d^2 v}{dj^2} \right] - \cancel{\delta e^{\frac{j\delta}{2}} \frac{dv}{dj}} - \frac{\delta^2}{2} e^{\frac{j\delta}{2}} v(j) = r_p^2 \delta^2 e^{2j\delta} [V(j) - E] v(j)e^{\frac{j\delta}{2}}$$

$$e^{\frac{j\delta}{2}} \frac{d^2 v}{dj^2} - \frac{\delta^2}{4} e^{\frac{j\delta}{2}} v(j) = r_p^2 \delta^2 e^{2j\delta} [V(j) - E] v(j)e^{\frac{j\delta}{2}}$$

(18)

$$\therefore \boxed{\frac{d^2 v}{dj^2} = r_p^2 \delta^2 e^{2j\delta} \left[V(j) - E + \frac{1}{4r_p^2 e^{2j\delta}} \right] v(j)} \text{ Q.E.D.}$$

(19)

5.1 - e)

Resolvemos Kohn-Sham, usando $u(j) = v(j)e^{\frac{j\delta}{2}}$,

$$\frac{d^2 u(r)}{dr^2} = -2 [\epsilon - V_{ef}^\sigma(r)] u(r)$$

$$\frac{d^2 v}{dj^2} = \left\{ \frac{\delta^2}{4} - 2r_p^2 \delta^2 e^{2j\delta} [\epsilon - V_{ef}^\sigma(j)] \right\} v(j),$$

(20)

aplicando o método de Numerov (integrando de j_{max} a 0 e tomando $h = 1$),

$$v(j-1) = \frac{2 \left[1 + \frac{5}{12} f(j) \right] v(j) - \left[1 - \frac{1}{12} f(j+1) \right] v(j+1)}{\left[1 - \frac{1}{12} f(j-1) \right]},$$

(21)

$$\text{sendo } f(j) = \frac{\delta^2}{4} - 2r_p^2 \delta^2 e^{2j\delta} [\epsilon - V_{ef}^\sigma(j)].$$
(22)

5.1 - e) Código

```
v(n) = u(n)*exp(-n*delta / 2)
v(n-1) = u(n-1)*exp(-(n-1)*delta / 2)
...
do j=1, n
    f(j) = (delta**2)/4 - 2*(rp**2)*(delta**2)*&
    &exp(2*j*delta)*(EigenValue-Potential(j))
end do

v = Numerov(v, f, 1._8, n)

do j=2, n-1
    u(n-j) = v(n-j)*exp((n-j)*delta / 2)
end do
```

5.1 - e) Resultados

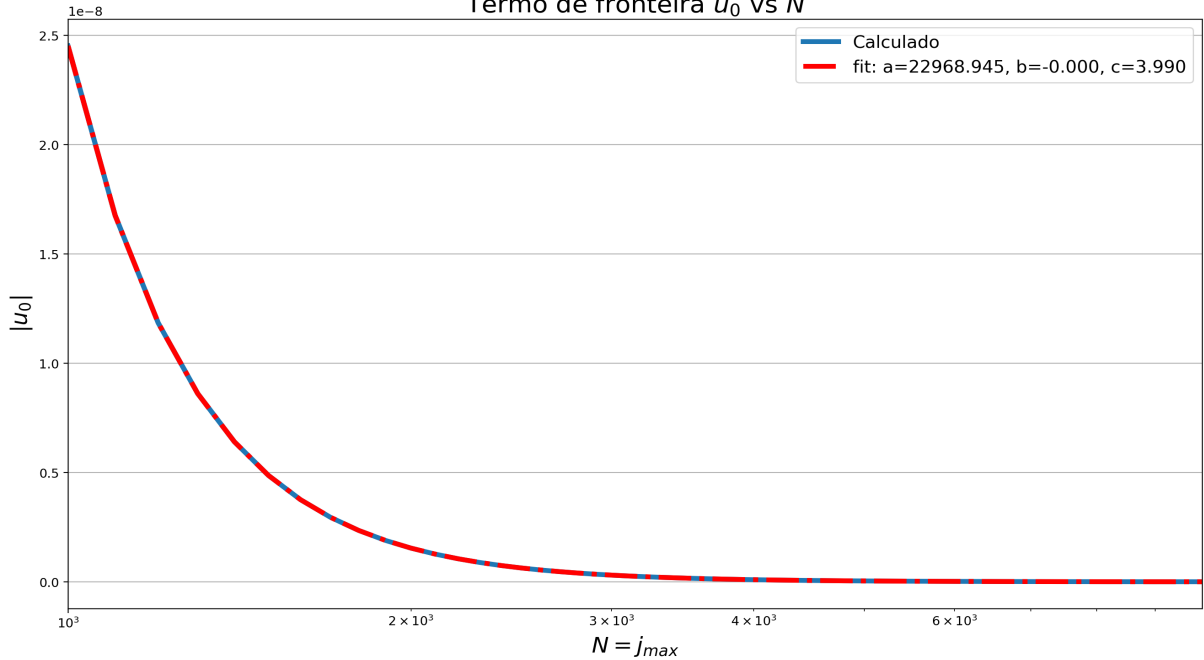
```
r_range = (/ 0., 50./), Eigenvalue_Range = (/ -5., 0./)
eigenvalue_tol = 0.00001, u0_tol= 0.001
h = 0.0001
j_max = 500000, delta = 0.0001
```

Malha	$u(0)$	Erro (abs) do autovalor	cpu_time (s)
Uniforme	$4,122 \cdot 10^{-6}$	$2,079 \cdot 10^{-4}$	0,18
Não-uniforme	$3,371 \cdot 10^{-6}$	$4,005 \cdot 10^{-5}$	0,76
Não-uniforme (Numerov)	$9,538 \cdot 10^{-7}$	$1,907 \cdot 10^{-6}$	0,34

5.1 - e) f2py - Erro de u_0 para autovalor exato $\epsilon = -0,5$

```
$ python -m numpy.f2py -c pydft.f95 -m pydft
...
from scipy.optimize import curve_fit
import pydft
...
def func(x, a, b, c):
    return a/(x**c) + b
...
u0[i] = np.abs(pydft.hydrogenatom(r_range, Eigenvalue_range,
    KS_int_max, Eigenvalue_tol, u0_tol,
    Uniform_Numerov, h, j_max[i], delta[i], write_data))
...
popt, pcov = curve_fit(func, j_max, u0)
```

Termo de fronteira u_0 vs N





[*https://github.com/HenriqueTortura/Density_Functional_Theory*](https://github.com/HenriqueTortura/Density_Functional_Theory)

- Funções *semi*-documentadas
- Mensagens de commits preguiçosas

Finocchi, F. (2011). *Density Functional Theory for Beginners*. Institut des NanoSciences de Paris (INSP), CNRS e University Pierre et Marie Curie.

URL: <https://www.semanticscholar.org/paper/Density-Functional-Theory-for-Beginners-Basic-and-Finocchi/db71416750ed288f0e1c8d3d821cbb2579517cd4>.

Martin, Richard M. (2004). *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press. DOI: [10.1017/CB09780511805769](https://doi.org/10.1017/CB09780511805769).

Thijssen, Jos (2007). *Computational Physics*. 2^a ed. Cambridge University Press. DOI: **10.1017/CB09781139171397**.