Solving Radial Kohn-Sham Equations Including l=1 Orbitals

In this script we implement KS-DFT for neutral atoms with $Z = 10$ (neon) and $Z = 12$ (magnesium). Assuming that in the ground state for $Z = 10$: $1s$, $2s$, and $2p$ shells are filled by 2, 2, and 6 electrons and for $Z = 12$: $1s$, $2s$, $2p$, and $3s$ shells are filled by 2, 2, 6, and 2 electrons respectively, so that ground state electron density is

spherically symmetric due to relation $\sum_{m=-\ell}^{\ell} Y_\ell^{m*}(\theta, \phi) Y_\ell^m(\theta, \phi) = \frac{2\ell + 1}{4\pi}$.

We implement numerically 2 separate Hamiltonians,

$$\ell = 0 \text{ sector: } \quad \hat{\mathscr{H}}_r u_i(r) = \lambda_i u_i(r), \quad \hat{\mathscr{H}}_r = -\frac{1}{2}\frac{\partial^2}{\partial r^2} + V_{\text{eff}}[\rho](r) = -\frac{1}{2}\frac{\partial^2}{\partial r^2} + V_{\text{ext}}(r) + V_{\text{H}}[\rho](r) + V_{\text{xc}}[\rho](r). \tag{1}$$

$$\ell = 1 \text{ sector: } \quad \hat{\mathscr{H}}_r \tilde{u}_i(r) = -\frac{1}{2}\frac{d^2\tilde{u}_i(r)}{dr^2} + \left[V_{\text{eff}}[\rho](r) + \frac{1}{2}\frac{\ell(\ell + 1)}{r^2}\right]\tilde{u}_i(r) = \tilde{\lambda}_i\tilde{u}_i(r). \tag{2}$$

Eq. (1) is for calculating s-orbital KS wave functions $\ell = 0$: $u_1, u_2, ...$ and their corresponding eigen values λ_1, λ_2,...(ordered in increasing order starting from the lowest eigen value). Hamiltonian (tilde variables) in Eq. (2) is for calculating p-orbital KS wave functions with $\ell = 1$: $\tilde{u}_1, \tilde{u}_2, ...$ and their eigen values $\tilde{\lambda}_1, \tilde{\lambda}_2, ...$ Then we calculate total ground state energy for neon as:

$$E_0 = 2\lambda_1 + 2\lambda_2 + 6\tilde{\lambda}_1 - E_{\text{H}} - \int V_{\text{xc}}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + E_{\text{xc}}$$

and for magnesium as:

$$E_0 = 2\lambda_1 + 2\lambda_2 + 6\tilde{\lambda}_1 + 2\lambda_3 - E_{\text{H}} - \int V_{\text{xc}}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + E_{\text{xc}}.$$

To calculate total electron density we define for neon: Nband=3 and `S.occ=[2 2 6]`, for magnesium: `Nband=4` and `S.occ=[2 2 6 2]`.

To help convergence of algorithm it is essential to implement density mixing schemes, hence select nonzero value of beta in line 105, say `beta=0.2`.

We start with selecting parameters for simulating ground state energy and electron density using our radial KS-DFT numerical solver that is included at the end of this script.

First we have to select the atom index for neon $Z = 10$ and for magnesium $Z = 12$. Note, this code is applicable only for neon or magnesium atoms.

```
Z = 10;
```

Next, we set up following parameters:

- The maximum radial length R_{max} - simulation "box" size in Bohr radius units.

- Number of grid points in radial direction N_d. Simulations on system sizes of up to $N_d = 1000$ is possible for `tol` $= 10^{-4}$.
- Order of finite difference approximation `FDn`. We use central finite differences for discrete approximations of derivatives.
- Convergence tolerance: setting a threshold for the difference between total energies obtained in the current and previous steps of iteration. Once the difference becomes lower than tolerance, iteration stops.

```
R_MAX = 30;
Nd=2000;
FDn = 4;
tol = 1e-4;
```

Now we run our radial Kohn-Sham DFT code.

```
S_DFT = SCF(init_DFT(Z,R_MAX,Nd,tol,FDn));
```

```
Total energy in 1 step -114.229788 Ha
Total energy in 2 step -143.306507 Ha
Total energy in 3 step -126.500752 Ha
Total energy in 4 step -126.074605 Ha
Total energy in 5 step -125.119601 Ha
Total energy in 6 step -125.383227 Ha
Total energy in 7 step -125.112566 Ha
Total energy in 8 step -125.269522 Ha
Total energy in 9 step -125.155676 Ha
Total energy in 10 step -125.230973 Ha
Total energy in 11 step -125.179210 Ha
Total energy in 12 step -125.214222 Ha
Total energy in 13 step -125.190372 Ha
Total energy in 14 step -125.206568 Ha
Total energy in 15 step -125.195554 Ha
Total energy in 16 step -125.203038 Ha
Total energy in 17 step -125.197951 Ha
Total energy in 18 step -125.201409 Ha
Total energy in 19 step -125.199058 Ha
Total energy in 20 step -125.200656 Ha
Total energy in 21 step -125.199570 Ha
Total energy in 22 step -125.200308 Ha
Total energy in 23 step -125.199806 Ha
Total energy in 24 step -125.200147 Ha
Total energy in 25 step -125.199915 Ha
Total energy in 26 step -125.200073 Ha
Total energy in 27 step -125.199966 Ha
Total energy in 28 step -125.200039 Ha
Finished SCF in 28 steps!

Energy decomposition
Ek   =   99.361991 Ha
Eh   =   63.004750 Ha
Exc  = -11.471467 Ha
Eext = -295.419119 Ha
Etot = -125.200039 Ha, -3406.869329 eV
```

Next, we plot the ground state radial electron density $4\pi\rho(r)r^2 = 4\pi\rho_\uparrow(r)r^2 + 4\pi\rho_\downarrow(r)r^2$ for neon atom.

```
figure
hold on
plot([0;S_DFT.r],[0;4*pi*S_DFT.rho.*S_DFT.r.^2])  % adding r=0 point
    legend("KS-DFT $\rho$",'Interpreter','latex')
hold off

xlabel("$r$ (Bohr)",'Interpreter','latex')
ylabel("$4\pi \rho(r) r^2$",'Interpreter','latex')
xlim([0 6])
title("Radial electron density",'Interpreter','latex')
subtitle(['$Z=$',num2str(Z)],'Interpreter','latex')
box on
```


Radial electron density
$Z = 10$
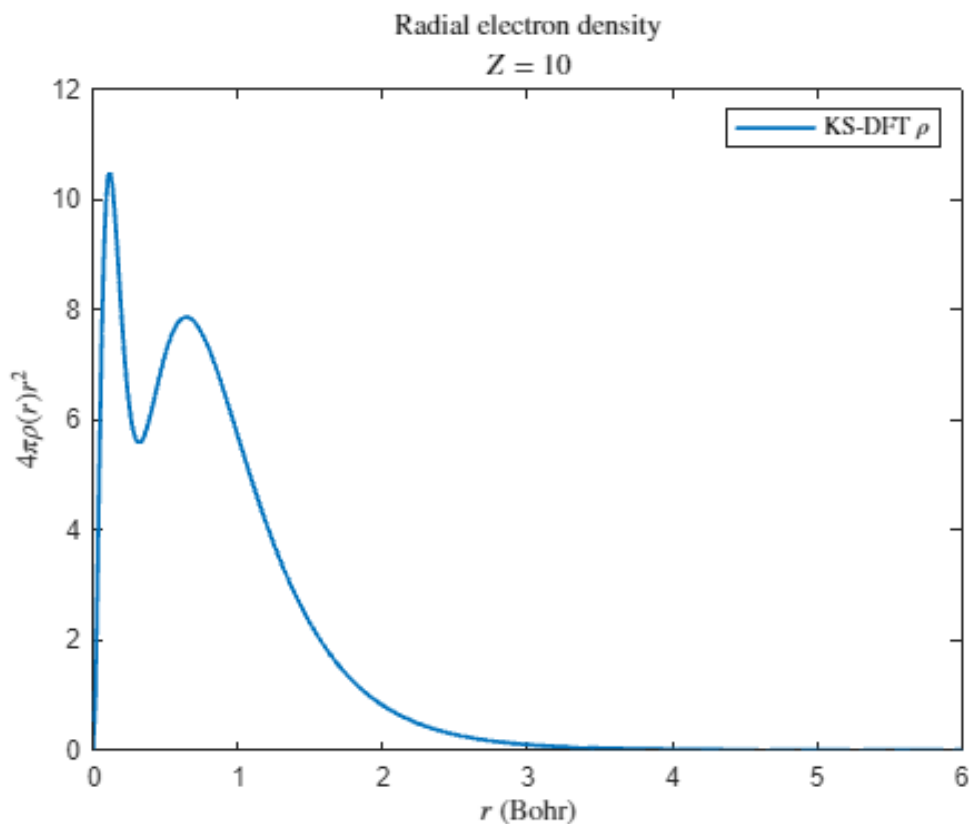
Next, we plot the converged Hartree potential $V_{\mathrm{H}}[\rho_{converged}](r)$ and exchange-correlation potential $V_{\mathrm{xc}}[\rho_{converged}](r)$ (if any) and external potential $V_{\mathrm{ext}}(r)$.
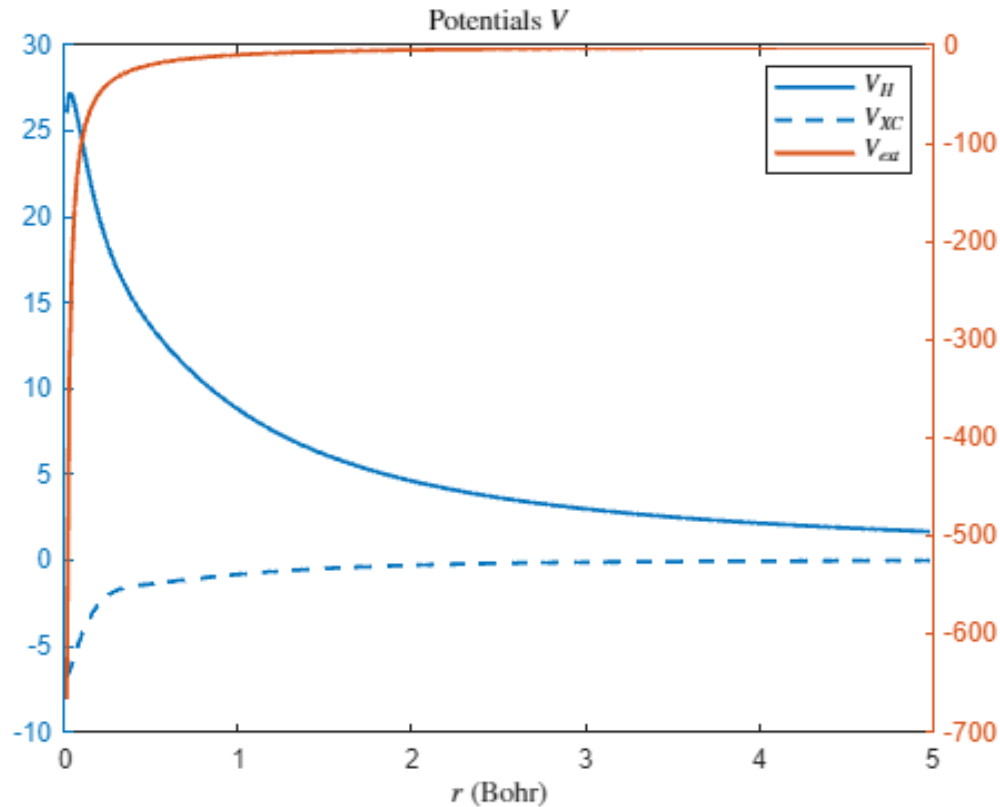
```
figure
yyaxis left
plot(S_DFT.r,S_DFT.Vh)
hold on
plot(S_DFT.r,S_DFT.Vx+S_DFT.Vc)
hold off
yyaxis right
```

```
plot(S_DFT.r,S_DFT.Vext)
legend("$V_H$","$V_{XC}$","$V_{ext}$",'Interpreter','latex')
xlabel("$r$ (Bohr)",'Interpreter','latex')
xlim([0 5])
title("Potentials $V$",'Interpreter','latex')
box on
hold off
```
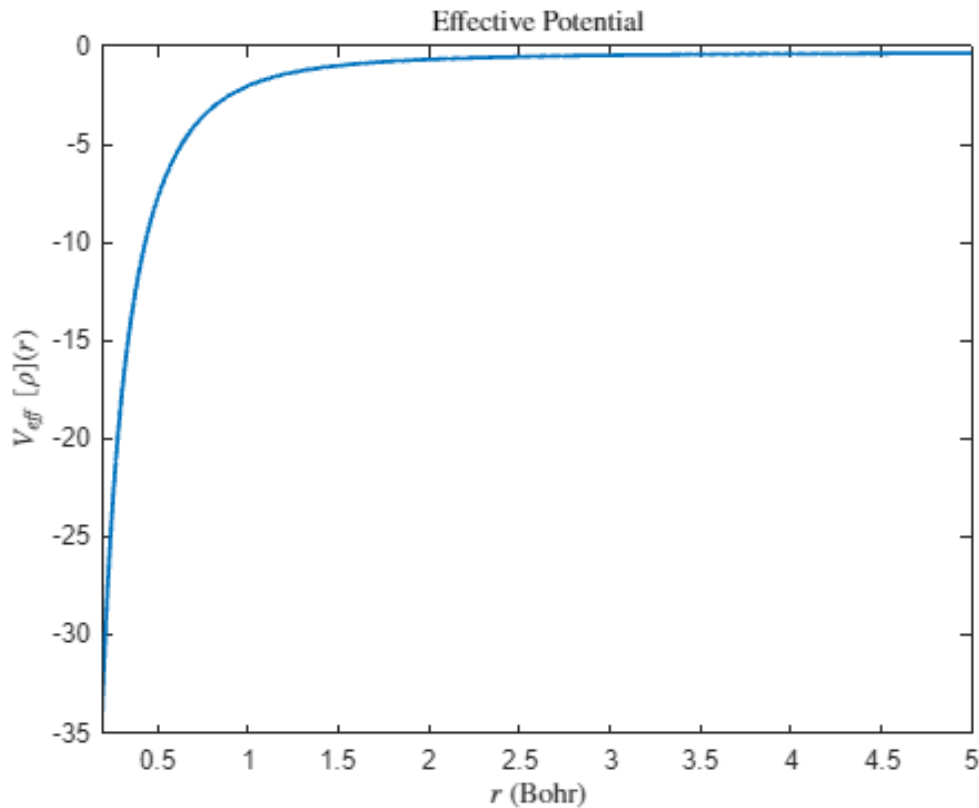

Potentials V

And lastly we plot the converged effective potential $V_{eff} = V_{ext} + V_H + V_{xc}$

```
figure
plot(S_DFT.r,S_DFT.Vh+S_DFT.Vx+S_DFT.Vc+ S_DFT.Vext)
xlim([0.2,5])
ylabel("$V_{eff}\,\,[\rho](r)$",'Interpreter','latex')
xlabel("$r$ (Bohr)",'Interpreter','latex')
title( " Effective Potential",'Interpreter','latex')
```

Effective Potential

Simple Numerical Solver for Radial Kohn-Sham DFT, Including $\ell = 1$ Filled Subshell

We implement two separate Hamiltonians as in Eq. (1) and Eq. (2), more precisely, actions of these two Hamiltonians on vectors.

```
function Hr_u = Hamiltonian(S,u)
    % Hamiltonian times vector routine for l=0 sector
    Hr_u = -1/2*S.Lap*u + S.Veff.*u;
    end

    function Hr_l_u = Hamiltonian_l(S,u)
    % Hamiltonian times vector routine for l=1 sector
    Hr_l_u = -1/2*S.Lap*u + (S.Veff+1./S.r.^2).*u;
end
```

Next we implement the effective $V_{\text{eff}}[\rho](r) = V_{\text{ext}}(r) + V_{\text{H}}[\rho](r) + V_{\text{xc}}[\rho](r)$. Note that, after discretizing the radial direction, all potentials are stored as N_d-dimensional columnar vectors, for example $V_{ext}(r) \rightarrow \{V_{ext}(r_1), V_{ext}(r_2), ..., V_{ext}(r_{N_d})\}$.

```
function S = compute_effective_potential(S,rho)
    % compute effective potential
```

5

```matlab
    S.Vh = compute_hartree_potential(S,rho);
    [S.Ex,S.Vx] = slater(rho);      % exchange contribution in energy and effective
 potential
    [S.Ec,S.Vc] = pz(rho);          % Perdew-Zunger correlation contribution
    % total effective potential
    S.Veff = S.Vext + S.Vh + S.Vx + S.Vc;
    % S.Vext is created during the initialization of S
end
```

Function implementing electron density.

```matlab
function rho = electron_density(S,u)
    R = u./S.r;
    rho = sum(R.^2.*S.occ,2)/(4*pi);
end
```

Hartree potential

```matlab
function Vh = compute_hartree_potential(S,rho)
    % Computes Hartree potential from solving the Poisson equation
    % phi''(r) = -4pi*r*rho(r), where Vh = phi/r

    phi = S.Lap\(-4*pi*rho.*S.r);
    Vh = phi./S.r;
end
```

Function `int3d` to calculate integral of a spherically symmetric function f over $\mathbf{R}^3$ and approximate integral with a discrete sum over the 1D radial grid.

```matlab
function intf = int3d(S,f)

    intf = 4*pi*S.h*sum(f.*S.r.^2);
end
```

Function calculating total energy and different contributions to it.

```matlab
function S = energy(S)
    % compute all energies
    S.Ek = sum(sum((-1/2*S.Lap*S.u).*S.u).*S.occ)*S.h;
    S.Eext = int3d(S,S.Vext.*S.rho);
    S.Eh = 1/2*int3d(S,S.Vh.*S.rho);
    S.Exc = int3d(S,(S.Ex+S.Ec).*S.rho);
    S.Exc_dc = int3d(S,(S.Vx+S.Vc).*S.rho);
    S.Etot = sum(S.occ.*S.Eband') - S.Eh - S.Exc_dc + S.Exc;
end
```

The normalization condition on electron density becomes

$$4\pi \int \rho(r)r^2 dr = Z$$

```matlab
function u = normalize(S,u)
    % rescale radial wave function

    scale = S.h*sum(u.^2,1);   % S.h= Delta r
    u = u ./ sqrt(scale);
end
```

Self-consistent field method for solving the radial Kohn-Sham eqution

```matlab
function S = SCF(S)
```

Step 1: initializing the electron density by using 1s wave function.

```matlab
    % initial 1s wave function and electron density
    R0 = 2*S.Z^(3/2)*exp(-S.Z*S.r);

    S.u = normalize(S,R0.* S.r);      % u=r*R
    S.rho = electron_density(S,S.u);
    S.rho0 = S.rho;

    % SCF parameters
    Etot_prev = 1;
    S.Etot = 0;
    step = 0;
    maxstep = 3000;
```

Here is the control of the SCF by using the difference between total energies, obtained in two consequtive iteration steps, as convergence criteria (this is **Step 5**).

```matlab
    while abs(Etot_prev-S.Etot)>S.tol && step < maxstep
        Etot_prev = S.Etot;
```

Step 2: Calculate effective potential for the current Hamiltonian.

```matlab
        S = compute_effective_potential(S,S.rho);
```

Step 3: Solve Kohn-Sham equation.

```matlab
        [S.u,S.Eband] = eigSolve(S);
```

Step 4: Update electron density $\rho(r)$ and the total energy.

To avoid large charge redistribution from one iteration to the next and help algorithm to converge, we mix new and old densities at each iteration.

```matlab
        beta=0.3;              % try out beta=0.2, 0.5, and beta=0.9;
        S.rho = (1-beta)*electron_density(S,S.u)  +beta *S.rho; % S.rho is updated
```

```
            S = energy(S);
            step = step + 1;
            fprintf("Total energy in %d step %.6f Ha\n", step, S.Etot);
        end
        fprintf('Finished SCF in %d steps!\n', step);
        fprintf("\nEnergy decomposition \n");
        fprintf("Ek   = %10.6f Ha\n", S.Ek);
        fprintf("Eh   = %10.6f Ha\n", S.Eh);
        fprintf("Exc  = %10.6f Ha\n", S.Exc);
        fprintf("Eext = %10.6f Ha\n", S.Eext);
        fprintf("Etot = %10.6f Ha, %10.6f eV\n", S.Etot, S.Etot * 27.2114079527);
end
```

Kohn-Sham equation is an eigenvalue problem. Since we have decided to implement instead of Hamilton matrix, action of Hamiltonian on N_d-dimensional vector $\vec{u}_j$, we define function handle Hfun that eigs function will accept instead of Hamilton matrix.

```
function [u,Eband] = eigSolve(S)
    % solve eigenvalue problem and normalize wave function

    opts = struct(maxit= 5000, tol=S.tol,isreal=1);
    Hfun = @(x) Hamiltonian(S,x);
    Hfun_l = @(x) Hamiltonian_l(S,x);

    [u,Eband] = eigs(Hfun,S.Nd,S.Nband,'smallestreal',opts);
    [u_l,Eband_l] = eigs(Hfun_l,S.Nd,S.Nband,'smallestreal',opts);
    Eband=diag(Eband);
    Eband_l=diag(Eband_l);

    if S.Z==12
        Eband(4)=Eband(3);
        u(:,4)=u(:,3);
    end

    Eband(3)=Eband_l(1);
    u(:,3)=u_l(:,1);

    u = normalize(S,u);
end
```

init_DFT is the initialization function to set up parameters for the Kohn-Sham problem.

```
function S = init_DFT(Z,R_MAX,Nd,tol,FDn)
    % Structure S picks up 11 fields after initialization
    S = struct();
    S.Z = Z;
    S.Nd = Nd;
    S.h = R_MAX/Nd;   %
    S.r = linspace(S.h,R_MAX,Nd)';
```

```
        S.FDn = FDn;
        S.tol = tol;
        S.Etot = 0;
        S.Vext = - S.Z./ S.r;
        S.Lap = Laplacian(S.FDn,S.Nd,S.h);
```

For neon we implement occupation of $1s$, $2s$, and $2p$ shells by 2, 2, and 6 electrons respectively.

For magnesium we implement occupation of $1s$, $2s$, $2p$, and $3s$ shells by 2, 2, 6, and 2 electrons respectively.

```
    if S.Z==10         % case of neon
        S.Nband=3;
        S.occ=[2 2 6];
    elseif S.Z==12     % case of magnesium
        S.Nband=4;
        S.occ=[2 2 6 2];
    end
 end
```

Below are pair of functions (`slater` and `pz`) for implementing LDA exchange and LDA-PZ correlation functional and potential.

```
 %%%%%% exchange correlation functions %%%%%%

 function [ex,vx] = slater(rho)
     % slater exchange

     C2 = 0.73855876638202; % 3/4*(3/pi)^(1/3)
     C3 = 0.9847450218427;  % (3/pi)^(1/3)
     ex = - C2 * rho.^(1./3.);
     vx = - C3 * rho.^(1./3.);
 end

 function [ec,vc] = pz(rho)
     % pz correlation: Perdew-Zunger implementation of LDA correlation energy and
 potential

     A = 0.0311;
     B = -0.048;
     C = 0.002;
     D = -0.0116;
     gamma1 = -0.1423;
     beta1 = 1.0529;
     beta2 = 0.3334;

     % computation
     ec = zeros(size(rho,1),1);
     vc = zeros(size(rho,1),1);
     rs = (0.75./(pi*rho)).^(1.0/3.0) ;
     islt1 = (rs < 1.0);
```

```matlab
    lnrs = log(rs(islt1));
    sqrtrs = sqrt(rs(~islt1));
    ec(islt1) = A * lnrs + B + C * rs(islt1) .* lnrs + D * rs(islt1);
    ox = 1.0 + beta1*sqrtrs + beta2*rs(~islt1);
    ec(~islt1) = gamma1 ./ ox;
    vc(islt1) = lnrs.*(A + (2.0/3.0)*C*rs(islt1)) + (B-(1.0/3.0)*A) +
(1.0/3.0)*(2.0*D-C)* rs(islt1);
    vc(~islt1) = ec(~islt1) .* (1 + (7.0/6.0)*beta1*sqrtrs +
(4.0/3.0)*beta2*rs(~islt1)) ./ ox;
end
```

```matlab
%%%%%% finite difference functions %%%%%%

function Lap = Laplacian(FDn,Nd,h)

    w2 = w2_c(FDn);
    e = ones(Nd,1)/h^2;
    Lap = spdiags(w2(1)*e,0,Nd,Nd);
    for i = 1:FDn
        Lap = Lap + spdiags(w2(i+1)*e,i,Nd,Nd);
        Lap = Lap + spdiags(w2(i+1)*e,-i,Nd,Nd);
    end
    end


    function w2 = w2_c(FDn)

    w2 = zeros(1,FDn+1) ;
    for k=1:FDn
        w2(k+1) = (2*(-1)^(k+1))*(factorial(FDn)^2)/...
            (k*k*factorial(FDn-k)*factorial(FDn+k));
        w2(1) = w2(1)-2*(1/(k*k));
    end
    end
end
```

For neon we obtain $E_0 \simeq -3407eV$ for $N_d = 2000$, $R_{max} = 30$ and `tol`$= 10^{-4}$, whereas result when only s-orbitals are retained with same simulation parameters is $E_0 \simeq -3052eV$.

- reported in the literature ground state energy, sum of all of ionization energies of neon is: $E_0 \simeq -3512eV$
- the result of our orbital-free implementation of DFT for neon is: $E_0 \simeq -3428eV$ for same simulation parameters as here and $E_0 \simeq -3507eV$ for $N_d = 8000$.

For magnesium we get $E_0 \simeq -5259eV$ for $N_d = 2000$, $R_{max} = 30$, and `tol`$= 10^{-4}$, whereas result when retaining only s-orbitals is $E_0 \simeq -4525eV$ for same simulation parameters.

- reported result in literature is $E_0 \simeq -5451 eV$
- the result of our orbital-free implementation of DFT for magnesium is $E_0 \simeq -5269 eV$ for same simulation parameters as here and $E_0 \simeq -5409 eV$ for $N_d = 8000$.