1. Preface

The pseudopotential approach is based on the observation that most physical and chemical properties of atoms are determined by the structure and dynamics of the atomic valence states. It exploits this by removing the core electrons as well as by replacing them and the strong ionic potential by a weaker pseudopotential that acts on a set of nodeless pseudo-wavefunctions rather than the true valence wave functions. Using the pseudopotential approximation, it will be easy to handle the valence electrons of the system and to reduce the dimension of the Hamiltonian matrix. The construction of the pseudopotential is described below.

First we compute the total ground state energy of a many-electron system. For convenience, we use the atomic unit with $\hbar = m = e = 1$. It is a self-consistent calculation. The procedure requires an initial guess for the electronic density, from which the Hartree potential and the exchange-correlation potential can be calculated. After self-consistent calculations, we can obtain the Kohn-Sham eigenstates of the ground state wave functions of atoms. These eigenstates will normally generate the ground state charge density. Our computation results are close to nonrelativistic experimental results.

In the total energy computation of many-body electron system, it is desired to reduce the number of basis in the momentum space and to reduce the drastic charge near the nucleus. By an equivalent one with only the valence orbital and the corresponding pseudopotential, it is easy to replace the whole system. Having the ground state density, we can construct the pseudopotential with some guiding criteria. In the end, we will draw the pseudo-wave functions and wave functions of atoms.

2. The Premise of Physics

In last several decades, the algorithmic advance and the development of high-speed supercomputer have made Ab Initio quantum mechanical simulations possible for a wide range of physical systems. We will introduce how to compute the total ground state energy of atoms.

2.1 The Schrödinger equation

The equation has been developed (1926) by the Austrian physicist Erwin Schrödinger. For a single particle in three dimensions, the equation is:

$$i\hbar\frac{\partial}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\Psi(\mathbf{r},t) + V(\mathbf{r})\Psi(\mathbf{r},t)\,,$$

where $\Psi(\mathbf{r},t)$ is the wave function, which is the amplitude for the particle to have a given position $\mathbf{r}$ at any given time t, $\hbar$ is the value of the reduced Planck constant, m is the mass of the particle, and $V(\mathbf{r})$ is the potential. This equation is a time-dependent equation. By introducing the separation of variables, $\Psi(\mathbf{r},t)$ can be written as

$$\Psi(\mathbf{r},t) = \psi(\mathbf{r})f(t)\,.$$

and we obtain

$$\frac{i\hbar}{f}\frac{\partial f}{\partial t}=\left[-\frac{\hbar^2}{2m}\nabla^2\psi+V(\mathbf{r})\psi\right]\frac{1}{\psi}$$

In this equation, the left-hand side has only a variable of t while the right-hand side has only a variable of $\mathbf{r}$. Let both sides be equal to the same constant, called E (orbital energy). Then, we can obtain two equations:

$$-\frac{\hbar^2}{2m}\nabla^2\psi(\mathbf{r})+V(\mathbf{r})\psi(\mathbf{r}) \quad =E\psi(\mathbf{r}) \tag{1.1}$$

$$i\hbar\frac{\partial f(t)}{\partial t}=Ef(t) \tag{1.2}$$

Equation (1.1) is called the time-independent Schrödinger equation and can be written as:

$$H\psi(\mathbf{r})=E\psi(\mathbf{r}),$$

where H=$-\frac{\hbar^2}{2m}\nabla^2+V(\mathbf{r})$, called the Hamiltonian operator.

Evidently the time-dependent part is a pure phase factor. Therefore only dealing with the time-independent Schrödinger equation, one can get the total energy when the particle is in a static potential.

2.2 Density Functional theory and the Kohn-Sham equation

In principle, density functional theory (DFT) is an exact description of the many-body wavefunction, while in practice it spurs on varied types of daring approximations. The starting point of the theory is the observation of Hohenberg and Kohn (1964) that electronic density contains in principle all the information contained in a many-electron wave function. DFT was put on a firm theoretical footing by the two Hohenberg-Kohn theorems (H-K).

The first H-K theorem proved that the ground state properties of a many-electron system are uniquely determined by an electronic density. We can say that the density completely determines the many-body problem.

The second H-K theorem defines energy functional for the system and proves that the correct ground state electronic density minimizes this energy functional. The minimum value of the total energy functional is the ground state energy.

The total energy E of the system as a functional of the charge density n ($\mathbf{r}$) is written:

$$E[n] = T[n] + \int V_{ext}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} + \frac{1}{2}\int V_H[\mathbf{r}] n(\mathbf{r}) d\mathbf{r} + E_{xc}[n], \qquad (2.2)$$

where T is the kinetic energy of the system, V_{ext} is the external potential (e.g. the Coulomb potential due to the nuclear charges), V_H is the electron-electron Coulomb potential, E_{xc} is the exchange-correlation functional.

Only the minimum value of the total energy functional E[n] has an important physical meaning. We can obtain the total ground state energy by minimizing the total energy functional E[n], and the density that yields this minimum value is the ground state density. In order to minimize the total energy functional (2.2), we use Lagrange multiplier. The method of Lagrange multipliers provides a strategy for finding the maximum/minimum of a function subject to constraints.

$$\frac{\partial}{\partial \psi_i^*(\mathbf{r})}\left(E[n]-\varepsilon_i\left(N-\int n(\mathbf{r})d\mathbf{r}\right)\right)=0 \tag{2.3}$$

where ψ_i is the wave function of the state i, ε_i is the Lagrange multiplier and $\int n(\mathbf{r})d\mathbf{r}=N$ is the constraint. By equation (2.3), we can obtain the Kohn-Sham equation:

$$\left(-\frac{\hbar^2}{2m}\nabla^2+V_{ext}(\mathbf{r})+V_{XC}(\mathbf{r})+V_H(\mathbf{r})\right)=\varepsilon_i\psi_i(\mathbf{r}) \quad , \tag{2.4}$$

where ε_i is the Lagrange multiplier and may be interpreted as the orbital energy of the state represented by $\psi_i(\mathbf{r})$, V_{xc} is the exchange-correlation potential, and V_H is electron-electron Coulomb potential. The Kohn-Sham equation (2.4) represents a mapping of the interacting many-electron system onto a system of non-interacting electrons moving in an effective potential due to all the other electrons. The set of wave functions ψ_i are given by the ab initio self-consistent solutions to the Kohn-Sham equation (2.4) so that the occupied electronic states generate a charge density that produces the electronic potential that was used to construct the equations.

2.3 The Local density approximation

The Hohenberg-Kohn theorem provides some motivation for using approximate methods to describe the exchange-correlation energy as a function of the electron density. The local density approximation (LDA) is the simplest method of describing the exchange-correlation energy of an electron system. In LDA, the exchange-correlation energy of an electron system is constructed by assuming that the exchange-correlation energy $\varepsilon_{xc}(\mathbf{r})$ per electron at a point $\mathbf{r}$ in the electron gas, it is equal to the exchange-correlation energy per electron in a homogeneous electron gas that has the same density as the electron gas at point $\mathbf{r}$. Thus,

$$E_{xc}[n(\mathbf{r})] = \int \varepsilon_{xc}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} \tag{2.5}$$

and by [1], we have:

$$\frac{\delta E_{xc}\left[n(\mathbf{r})\right]}{\delta n(\mathbf{r})} = \frac{\partial\left[n(\mathbf{r})\varepsilon_{xc}(\mathbf{r})\right]}{\partial n(\mathbf{r})} \tag{2.6}$$

The LDA assumes that the exchange-correlation energy functional is purely local. In principle, it ignores correlations to the exchange-correlation energy at a point $\mathbf{r}$ due to nearby inhomogeneities in the electron density. Considering the inexact nature of the approximation, it is remarkable that calculations performed using the LDA have been so successful.

2.4 Pseudopotential

It is well known that most physical properties of solids are dependent on the valence electrons to a much greater extent than on the core electrons. Only the valence electrons play important roles in the calculation of more complex system such as molecules, clusters or band structures. The orbitals in the inner-core remain mostly unperturbed. It is desirable to replace the whole system by an equivalent one with only the valence orbital and the corresponding pseudopotential. The coulomb interaction of the electrons is the most time-consuming in real calculation and will become trivial in the momentum space. To reduce the number of the basis in the momentum space, it is desirable to reduce the drastic charge near the nucleus for the pseudopotential.

Following Hamann, Schluter and Chiang, the norm-conserving pseudopotential is constructed with the following guiding criteria:

1. Real and pseudo valence eigenvalues agree.
2. Real and pseudo atomic wave functions agree beyond a chosen "core radius" r_c.
3. The integrals from 0 to r of the real and pseudo charge densities agree for r> r_c for each valence state (norm conservation).
4. The logarithmic derivatives of the real and pseudo wave function and their first energy derivatives agree for r > r_c.

Properties (3) and (4) are important for the pseudopotential to have optimum transferability among a variety of chemical environments in self-consistent calculations in which the pseudo charge density is treated as a real physical object. Explicitly, the norm-conserving pseudo potential are constructed for each angular momentum *l*.

3. Process of study

3.1 The expression of the potential of the Kohn-Sham equation

In this note, we use the atomic unit with $\hbar = m = e = 1$. The ground state energy of many-electron system is express as a functional E[n] (see(2.2)) of the electronic density n ($\mathbf{r}$):

$$n(\mathbf{r}) = \sum_{i \in occupied} f_i \,|\, \psi(\mathbf{r}) \,|^2 , \tag{3.1}$$

where f_i are the occupation numbers of the orbital denoted by *i*. Because we use atomic unit, the Kohn-Sham equation (2.4) becomes:

$$\left(-\frac{1}{2}\nabla^2 + V_{ext}(\mathbf{r}) + V_{XC}(\mathbf{r}) + V_H(\mathbf{r}) \right) \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r}) \tag{3.2}$$

Here $V_{ext}(\mathbf{r})$ is the external potential that is written:

$$V_{ext}(\mathbf{r}) = -\frac{z}{r} , \tag{3.3}$$

where z is an atomic number(e.g. Mg z=12).

The exchange-correlation potential V_{XC} is given by:

$$V_{xc}(\mathbf{r}) = \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})} \tag{3.4}$$

In LDA, by the equation (2.5), we can get further:

$$V_{xc}(\mathbf{r}) = \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})} = \varepsilon_{xc}(\mathbf{r}) + n(\mathbf{r}) \frac{d\varepsilon_{xc}}{dn} \tag{3.5}$$

The electron-electron Coulomb potential V_H is written:

$$V_H(\mathbf{r}) = \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \tag{3.6}$$

Thus, the effective potential V_{eff} is written:

$$V_{eff}(\mathbf{r}) = V_{ext}(\mathbf{r}) + V_H(\mathbf{r}) + V_{xc}(\mathbf{r}) \tag{3.7}$$

3.2 The simplification of the effective potential

It is convenient to employ the spherical coordinates $\{r,\theta,\phi\}$. The eigenfunctions are written as:

$$\psi_i(\mathbf{r}) = R_{nl}(r)Y_l^m(\theta,\phi) \tag{3.8}$$

The electronic orbitals are represented by the index i=$\{n,l,m\}$, the main quantum number n, the angular momentum quantum number *l*, and the magnetic quantum number m respectively. Let us first consider the closed-shell system for example Ne atom with z=10. The electronic shells $1s^2, 2s^2, 2p^6$ are all fully occupied. For such a closed-shell system, the electronic density $n(\mathbf{r})$ is spherical symmetric:

$$n(\mathbf{r}) = n(r)$$

due to the identity by [2]

$$\sum_{m=-l}^{l} \left|Y_l^m\right|^2 = \frac{2l+1}{4\pi}$$

Hence the external potential $V_{ext}(n(\mathbf{r}))$ and the exchange-correlation potential $V_{xc}(n(\mathbf{r}))$ are manifestly spherical symmetric. Further more, using the addition theorem of angular momentum:

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{l=0}^{\infty} \frac{r_<^l}{r_>^{l+1}} p_l(\cos\theta_{\mathbf{r},\mathbf{r}'}) = \sum_{l=0}^{\infty} \frac{r_<^l}{r_>^{l+1}} \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_l^m(\theta,\phi) Y_l^{m*}(\theta',\phi'),$$

where $r_< = \min(\mathbf{r},\mathbf{r}')$,and $r_> = \max(\mathbf{r},\mathbf{r}')$.

We have

$$V_H(\mathbf{r}) = \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$$
$$= \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_l^m(\theta,\phi) \iiint r'^2 n(r') \frac{r_<^l}{r_>^{l+1}} dr' Y_l^{m*}(\theta',\phi') \sin\theta' d\theta' d\phi'$$
$$= 4\pi \int_0^{\infty} r'^2 n(r') \frac{1}{r_>} dr' \qquad (3.9)$$

Thus, the electron-electron Coulomb potential $V_H(n(\mathbf{r}))$ is also spherical symmetric.

The exchange-correlation energy per unit density $\varepsilon_{xc}(n)$ may be decomposed into a sum due to the exchange energy and the correlation energy.

$$\varepsilon_{xc}(n) = \varepsilon_x(n) + \varepsilon_c(n) \qquad (3.10)$$

They may be obtained from a system of uniform electron gas with the corresponding density. They are frequently expressed in terms of the radius r_s of a unit charge defined by

$$r_s = \left(\frac{3}{4\pi n}\right)^{1/3} \qquad (3.11)$$

The exchange term can be calculated exactly for a uniform electron gas, and is given by

$$\varepsilon_x(n) = -\frac{3}{4\pi r_s}\left(\frac{9\pi}{4n}\right)^{1/3} = -\frac{0.458}{r_s} \qquad (3.12)$$

In general, the magnitude of the correlation energy is much smaller than the exchange energy, and yet it is much harder to calculate. The correlation energy was calculated by Ceperley and Alder using Monte-Carlo simulation and was parameterized by Perdew and Zunger as:

$$\varepsilon_c = \begin{cases} -\gamma/(1+\beta_1\sqrt{r_s}+\beta_2 r_s) & \text{for } r_s \geq 1, \text{ low density,} \\ A\ln r_s + B + C r_s \ln r_s + D r_s & \text{for } r_s < 1, \text{ high density,} \end{cases} \qquad (3.13)$$

where the values of the constants are given by

$$\gamma = 0.1423; \quad \beta_1 = 1.0529; \quad \beta_2 = 0.3334$$

and

$$A = 0.0311;\quad B = -0.0480;\quad C = 0.0020;\quad D = -0.0116$$

By the equation (3.10) to (3.12), the exchange-correlation potential V_{xc} can be simplified as:

$$V_{xc}(r) = \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})} = \varepsilon_{xc}(\mathbf{r}) + n(\mathbf{r})\frac{d\varepsilon_{xc}}{dn}$$

$$= \varepsilon_{xc}(\mathbf{r}) + n(\mathbf{r})\frac{d\varepsilon_{x}}{dn} + n(\mathbf{r})\frac{d\varepsilon_{c}}{dn}$$

$$= \begin{cases} \varepsilon_{xc}(\mathbf{r}) + n(\mathbf{r})\left[0.458 r_s^{-2}\dfrac{dr_s}{dn}\right] + \\ \qquad n(\mathbf{r})\left[\gamma(1+\beta_1 r_s^{0.5} + \beta_2 r_s)^{-2}\cdot(0.5\beta_1 r_s^{-0.5}\dfrac{dr_s}{dn} + \beta_2\dfrac{dr_s}{dn})\right] & for\ r_s \ge 1, \\ \varepsilon_{xc}(\mathbf{r}) + n(\mathbf{r})\left[0.458 r_s^{-2}\dfrac{dr_s}{dn}\right] + \\ \qquad n(\mathbf{r})\left[A r_s^{-1}\dfrac{dr_s}{dn} + C\dfrac{dr_s}{dn}\ln r_s + C r_s r_s^{-1}\dfrac{dr_s}{dn} + D\dfrac{dr_s}{dn}\right] & for\ r_s < 1, \end{cases} \tag{3.14}$$

where $\frac{dr_s}{dn}$ is from the equation (3.11) and is written as:

$$\frac{dr_s}{dn} = -4(3)^{-2/3}\pi(4\pi n)^{-4/3}$$

Thus, the terms of the effective potential are obvious to calculate.

3.3 The simplification of the Kohn-Sham equation

Using the expression of the Laplace operator in the spherical coordinates:

$$\nabla^2\psi(\mathbf{r}) = \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial\psi}{\partial r}) + \frac{1}{r^2}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta\frac{\partial\psi}{\partial\theta}) + \frac{1}{\sin^2\theta}\frac{\partial^2\psi}{\partial^2\phi}\right) \tag{3.15}$$

And the identity:

$$\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta\frac{\partial}{\partial\theta}) + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial^2\phi}\right)Y_l^m = -l(l+1)Y_l^m \tag{3.16}$$

By the equation (3.15) and (3.16), the Kohn-Sham equation (3.2) now takes the form:

$$-\frac{1}{2}\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial R_{nl}}{\partial r})Y_l^m-\frac{1}{2}\frac{1}{r^2}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta\frac{\partial}{\partial\theta})+\frac{1}{\sin^2\theta}\frac{\partial^2}{\partial^2\phi}\right)Y_l^m R_{nl}+V_{eff}R_{nl}Y_l^m=\varepsilon_i R_{nl}Y_l^m$$

Using the equation (3.16), we have:

$$-\frac{1}{2}\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial R_{nl}}{\partial r})Y_l^m-\frac{1}{2}\frac{1}{r^2}[-l(l+1)]Y_l^m R_{nl}+V_{eff}R_{nl}Y_l^m=\varepsilon_i R_{nl}Y_l^m$$

Canceling the both sides Y_l^m, we obtain:

$$-\frac{1}{2}\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial R_{nl}}{\partial r})+\frac{l(l+1)}{2r^2}R_{nl}+V_{eff}R_{nl}=\varepsilon_i R_{nl}$$

Introducing

$$P_{nl}(r)=rR_{nl}(r)$$

and multiplying both sides by r ,then we can obtain:

$$-\frac{1}{2}\frac{d^2P_{nl}(r)}{d^2r}+\frac{l(l+1)}{2r^2}P_{nl}(r)+V_{eff}(r)P_{nl}(r)=\varepsilon_{nl}P_{nl} \qquad (3.17)$$

Here we know $P_{nl}(0)=0;\ P_{nl}(r)=0$ as $r\to\infty$.

3.4 The discretization of the Kohn-Sham equation

By the Taylor series, we can get the discretization of $\frac{d^2P_{nl}(r)}{d^2r}$ that is written as:

$$\frac{d^2P_{nl}(x_i)}{d^2r}\approx\frac{P_{nl}(x_{i+1})-2P_{nl}(x_i)+P_{nl}(x_{i-1})}{h^2}, \qquad (3.18)$$

where h is the distance of the x_i-x_{i-1} . Using the equation (3.18) substitute for (3.17), we can get:

$$-\frac{1}{2}\frac{P_{nl}(x_{i+1})-2P_{nl}(x_i)+P_{nl}(x_{i-1})}{h^2}+\frac{l(l+1)}{2x_i^2}P_{nl}(x_i)+V_{eff}(x_i)P_{nl}(x_i)=\varepsilon_{nl}P_{nl}(x_i)$$

It can be written as a tridiagonal symmetric matrix:

$$\begin{pmatrix} \left[\frac{1}{h^2}+\frac{l(l+1)}{2x_1{}^2}+V_{eff}(x_1)\right] & -\frac{1}{2h^2} & & & 0 \\ -\frac{1}{2h^2} & \ddots & \ddots & & \\ & \ddots & \ddots & & -\frac{1}{2h^2} \\ 0 & & & -\frac{1}{2h^2} & \left[\frac{1}{h^2}+\frac{l(l+1)}{2x_{n-1}{}^2}+V_{eff}(x_{n-1})\right] \end{pmatrix} \begin{pmatrix} P_{nl}(x_1) \\ P_{nl}(x_2) \\ \vdots \\ \vdots \\ P_{nl}(x_{n-1}) \end{pmatrix} = \varepsilon_{nl} \begin{pmatrix} P_{nl}(x_1) \\ P_{nl}(x_2) \\ \vdots \\ \vdots \\ P_{nl}(x_{n-1}) \end{pmatrix}$$

Introducing

$$A = \begin{pmatrix} \left[\frac{1}{h^2}+\frac{l(l+1)}{2x_1{}^2}+V_{eff}(x_1)\right] & -\frac{1}{2h^2} & & & 0 \\ -\frac{1}{2h^2} & \ddots & \ddots & & \\ & \ddots & \ddots & & -\frac{1}{2h^2} \\ 0 & & & -\frac{1}{2h^2} & \left[\frac{1}{h^2}+\frac{l(l+1)}{2x_{n-1}{}^2}+V_{eff}(x_{n-1})\right] \end{pmatrix}$$

and

$$x = \begin{pmatrix} P_{nl}(x_1) \\ P_{nl}(x_2) \\ \vdots \\ \vdots \\ P_{nl}(x_{n-1}) \end{pmatrix}; \quad \lambda = \varepsilon_{nl},$$

This equation may also be written as:

$$Ax = \lambda x \tag{3.19}$$

In this equation, we can solve the eigenvalues and the eigenvectors to get the orbital energy and the wave function $P_{nl}(r)$.

3.5 The procedure of Ab Initio self-consistent

Having the equation (3.19), we can start to calculate. The procedure of ab initio self-consistent is shown in the flow diagram in Fig. 1. The procedure requires an initial guess for the electronic charge density. From which the Hartree potential and the exchange-correlation potential can be calculated.

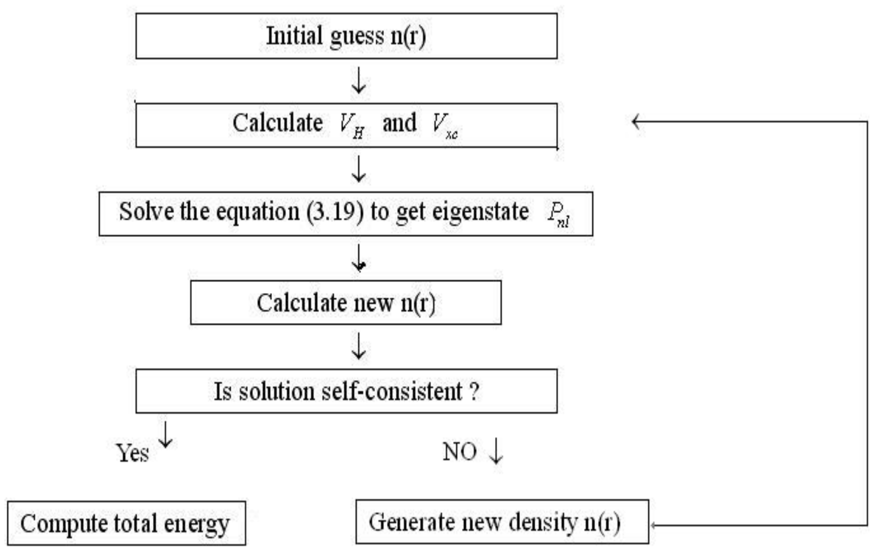


Fig. 1. Flow chart describing the computational procedure for the calculation of the ground state density of atoms.

In the end, we can obtain the ground state density of atoms to calculate total ground state energy.

3.6 The total ground state energy and simplification

From equation (2.2), the ground state energy functional is written as:

$$E[n] = T[n] + \int V_{ext}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} + \frac{1}{2}\int V_H[\mathbf{r}] n(\mathbf{r}) d\mathbf{r} + E_{xc}[n]$$

The first term on the right hand side is the kinetic energy functional that is expressed in terms of a system of noninteracting electrons.

$$T[n] = \sum_{i \in occupied} f_i \int \psi_i^*(\mathbf{r})(-\frac{\nabla^2}{2})\psi_i(\mathbf{r}) d\mathbf{r}$$

By multiplying both sides of equation (3.15) with $-\frac{1}{2}\psi_i^*(\mathbf{r})$, carrying out the integration, and then summing over all the occupied states *i*, the kinetic energy functional may be written as :

$$T[n] = \sum_{i \in occupied} f_i \int_0^\infty \left(-\frac{1}{2} P_{nl} \frac{d^2 P_{nl}}{d^2 r} + \frac{l(l+1)}{2r^2} P_{nl}^2 \right) dr \tag{3.20}$$

The second term on the right hand side of equation (2.2) is the external energy that can be simplified as:

$$\int V_{ext}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} = \iiint -\frac{z}{r} n(r) r^2 dr \sin\theta d\theta d\phi = -4\pi \int_0^\infty z r n(r) dr \tag{3.21}$$

The third term on the right hand side of equation (2.2) is the Hartree energy (electron-electron Coulomb energy) can be simplified as by using the addition theorem of angular momentum:

$$\begin{aligned} \frac{1}{2}\int V_H[\mathbf{r}] n(\mathbf{r}) d\mathbf{r} &= \frac{1}{2}\iint \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r} d\mathbf{r}' \\ &= \frac{1}{2}\iiint\iiint \frac{n(r) n(r') r^2 r'^2 \, dr dr' d\Omega d\Omega'}{|\mathbf{r}-\mathbf{r}'|} \\ &= \frac{1}{2}(4\pi)^2 \iint \frac{n(r) n(r') r^2 r'^2}{r_>} dr dr' \quad , \end{aligned} \tag{3.22}$$

where $d\Omega = \sin\theta d\theta d\phi$, $d\Omega' = \sin\theta' d\theta' d\phi'$.

In the subsection 3.5, we have gotten the ground state density n. Substituting the ground state density n for the energy functional equation (2.2), we can obtain the total ground state energy of atoms. In table 1 the resulting values of the ground state energy are displayed, for all atoms from z=1 to z=102, together with the corresponding values of $E_{tot}^{HF}(Z)$. The ratio $\left|\left(E[n]-E_{tot}^{HF}\right)/E_{tot}^{HF}\right|$ is shown in figure 2. As can be seen, the difference between the HF value and the value computed from equation (2.2) rarely exceeds 7%.

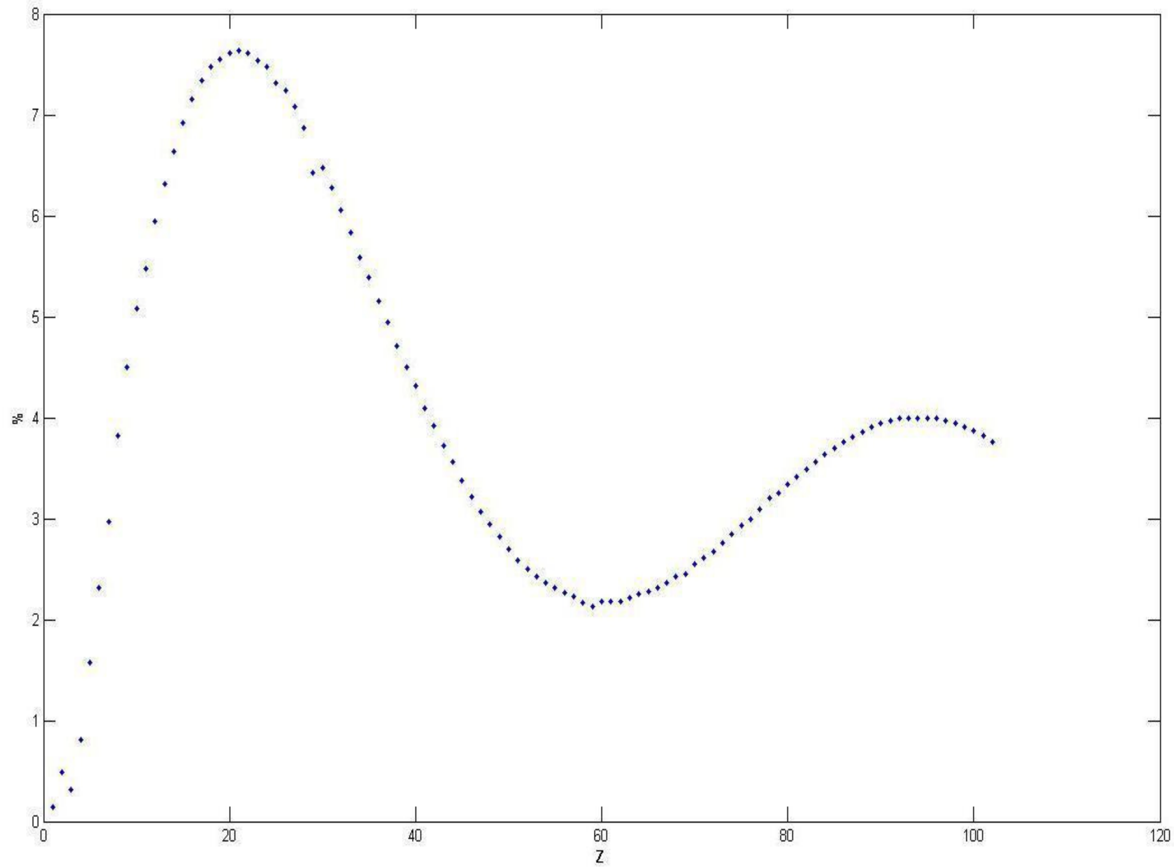


Figure 2.The ratio $\left|\left(E[n]-E_{tot}^{HF}\right)/E_{tot}^{HF}\right|$ plotted against Z. The values $E[n]$ and E_{tot}^{HF} are from table 1.

Table 1. Comparison of E[n],as computed from equation (2.2) of the text,with the Hartree-Fock value $E_{tot}^{HF}(Z)$ (from Froese-Fischer 1972, and Mann 1973).Energies are in atomic uints.

Z atom	$-E[n]$(a.u.)	$-E_{tot}^{HF}$(a.u.)	error (%)	Z atom	$-E[n]$(a.u.)	$-E_{tot}^{HF}$(a.u.)	error (%)	Z atom	$-E[n]$(a.u.)	$-E_{tot}^{HF}$(a.u.)	error (%)
1 H	0.5007	0.5	0.14	35 Br	2710.7	2572	5.39	69 Tm	13257	12940	2.45
2 He	2.8481	2.862	0.49	36 Kr	2893.9	2752	5.16	70 Yb	13732	13391	2.55
3 Li	7.4096	7.433	0.31	37 Rb	3083.4	2938	4.95	71 Lu	14213	13852	2.61
4 Be	14.688	14.57	0.81	38 Sr	3279.5	3132	4.71	72 Hf	14706	14323	2.67
5 B	24.918	24.53	1.58	39 Y	3482.1	3332	4.5	73 Ta	15209	14800	2.76
6 C	38.562	37.69	2.31	40 Zr	3691.5	3539	4.31	74 W	15722	15287	2.85
7 N	56.016	54.4	2.97	41 Nb	3907.7	3754	4.09	75 Re	16246	15784	2.93
8 O	77.665	74.81	3.82	42 Mo	4131	3975	3.92	76 Os	16781	16293	3
9 F	103.88	99.41	4.5	43 Tc	4361.5	4205	3.72	77 Ir	17326	16806	3.09
10 Ne	135.03	128.5	5.08	44 Ru	4599.3	4441	3.56	78 Pt	17888	17333	3.2
11 Na	170.76	161.9	5.47	45 Rh	4844.4	4686	3.38	79 Au	18447	17866	3.25
12 Mg	211.48	199.6	5.95	46 Pd	5097.1	4938	3.22	80 Hg	19023	18409	3.34
13 Al	257.17	241.9	6.31	47 Ag	5357.6	5198	3.07	81 Tl	19609	18962	3.41
14 Si	308.05	288.9	6.63	48 Cd	5625.7	5465	2.94	82 Pb	20205	19524	3.49
15 P	364.27	340.7	6.92	49 In	5901.6	5740	2.82	83 Bi	20811	20096	3.56
16 S	425.95	397.5	7.16	50 Sn	6185.4	6023	2.7	84 Po	21427	20676	3.63
17 Cl	493.21	459.5	7.34	51 Sb	6477.3	6314	2.59	85 At	22053	21267	3.7
18 Ar	566.17	526.8	7.47	52 Te	6777.4	6612	2.5	86 Rn	22689	21867	3.76
19 K	644.45	599.2	7.55	53 I	7085.9	6918	2.43	87 Fr	23333	22476	3.81
20 Ca	728.32	676.8	7.61	54 Xe	7402.9	7232	2.36	88 Ra	23986	23094	3.86
21 Sc	817.75	759.7	7.64	55 Cs	7728.2	7554	2.31	89 Ac	24649	23722	3.91
22 Ti	912.97	848.4	7.61	56 Ba	8062.1	7884	2.26	90 Th	25320	24360	3.94
23 V	1014	942.9	7.54	57 La	8404.6	8221	2.23	91 Pa	25999	25007	3.97
24 Cr	1121	1043	7.48	58 Ce	8753.1	8567	2.17	92 U	26687	25664	3.99
25 Mn	1234.2	1150	7.32	59 Pr	9111.3	8921	2.13	93 Np	27384	26331	4
26 Fe	1353.4	1262	7.24	60 Nd	9486.1	9284	2.18	94 Pu	28089	27008	4
27 Co	1478.8	1381	7.08	61 Pm	9865.6	9655	2.18	95 Am	28803	27696	4
28 Ni	1610.5	1507	6.87	62 Sm	10254	10035	2.18	96 Cm	29525	28392	3.99
29 Cu	1744.4	1639	6.43	63 Eu	10653	10423	2.21	97 Bk	30255	29100	3.97
30 Zn	1893.1	1778	6.47	64 Gd	11063	10820	2.25	98 Cf	30993	29817	3.94
31 Ga	2043.8	1923	6.28	65 Tb	11482	11226	2.28	99 Es	31739	30545	3.91
32 Ge	2200.8	2075	6.06	66 Dy	11911	11641	2.32	100 Fm	32493	31283	3.87
33 As	2364.3	2234	5.83	67 Ho	12350	12065	2.36	101 Md	33254	32031	3.82
34 Se	2534.2	2400	5.59	68 Er	12801	12498	2.42	102 No	34023	32790	3.76

3.7 Construction of Pseudopotential

Explicitly, the norm-conserving pseudopotential are constructed for each angular momentum l in the following steps: We fist solve for the radial part $P_{nl}(r)$ of the valence eigenfunctions ψ_i and the corresponding eigenvalues ε_{nl} by first carrying out an ab initio self-consistent of the figure 1. The radial parts of the valence states are given by the equation (3.17) or

$$-\frac{1}{2}\frac{d^2 P_{nl}(r)}{d^2 r}+V_l(r)P_{nl}(r)=\varepsilon_{nl}P_{nl}(r) \quad , \tag{3.23}$$

where

$$V_l(r)=\frac{l(l+1)}{2r^2}+V_{eff}(r) .$$

For each l, we choose a cutoff radius r_c, typically 0.5 to 1.0 times the radius r_m of the outermost peak of $P_{nl}(r)$. In table 2 to table 5, the choices of the cutoff radius r_c are displayed, for all atoms from z=1 to z=102.

Table 2. The choices of the cutoff radius r_c (Å) for each states of z=1 to z=25 atoms.

z	atom	electronic state	1s	2s	3s	4s	2p	3p	3d
1	H	$1s^1$	0.56						
2	He	$1s^2$	0.322						
3	Li	$1s^2 2s^1$	0.21	1.75					
4	Be	$1s^2 2s^2$	0.154	1.148					
5	B	$1s^2 2s^2 2p^1$	0.126	0.854			0.868		
6	C	$1s^2 2s^2 2p^2$	0.098	0.686			0.672		
7	N	$1s^2 2s^2 2p^3$	0.084	0.574			0.546		
8	O	$1s^2 2s^2 2p^4$	0.07	0.49			0.448		
9	F	$1s^2 2s^2 2p^5$	0.07	0.434			0.392		
10	Ne	$1s^2 2s^2 2p^6$	0.056	0.392			0.35		
11	Na	[Ne]$3s^1$	0.056	0.35	1.82		0.308		
12	Mg	[Ne]$3s^2$	0.056	0.308	1.428		0.266		
13	Al	[Ne]$3s^2 3p^1$	0.042	0.28	1.162		0.238	1.47	
14	Si	[Ne]$3s^2 3p^2$	0.042	0.266	1.008		0.21	1.19	
15	P	[Ne]$3s^2 3p^3$	0.042	0.238	0.896		0.196	1.022	
16	S	[Ne]$3s^2 3p^4$	0.042	0.224	0.798		0.182	0.896	
17	Cl	[Ne]$3s^2 3p^5$	0.042	0.21	0.728		0.168	0.798	
18	Ar	[Ne]$3s^2 3p^6$	0.028	0.196	0.672		0.154	0.714	
19	K	[Ar] $4s^1$	0.028	0.182	0.616	2.282	0.14	0.644	
20	Ca	[Ar] $4s^2$	0.028	0.168	0.574	1.904	0.14	0.588	
21	Sc	[Ar] $4s^2 3d^1$	0.028	0.168	0.532	1.764	0.126	0.532	0.616
22	Ti	[Ar] $4s^2 3d^2$	0.028	0.154	0.504	1.652	0.126	0.504	0.546
23	V	[Ar] $4s^2 3d^3$	0.028	0.154	0.476	1.582	0.112	0.476	0.49
24	Cr	[Ar] $4s^1 3d^5$	0.028	0.14	0.448	1.582	0.112	0.448	0.462
25	Mn	[Ar] $4s^2 3d^5$	0.028	0.14	0.42	1.442	0.098	0.42	0.42

Table 3. The choices of the cutoff radius r_c (Å) for each states of z=26 to z=50 atoms.

z	atom	electronic state	1s	2s	3s	4s	5s	2p	3p	4p	5p	3d	4d
26	Fe	[Ar] $4s^2 3d^6$	0.028	0.126	0.406	1.4		0.098	0.392			0.392	
27	Co	[Ar] $4s^2 3d^7$	0.028	0.126	0.392	1.344		0.098	0.378			0.378	
28	Ni	[Ar] $4s^2 3d^8$	0.028	0.126	0.378	1.302		0.098	0.364			0.35	
29	Cu	[Ar] $4s^1 3d^{10}$	0.028	0.112	0.364	1.372		0.084	0.35			0.378	
30	Zn	[Ar] $4s^2 3d^{10}$	0.028	0.112	0.35	1.232		0.084	0.322			0.322	
31	Ga	[Ar] $4s^2 3d^{10} 4p^1$	0.014	0.112	0.336	1.12		0.084	0.308	1.372		0.294	
32	Ge	[Ar] $4s^2 3d^{10} 4p^2$	0.014	0.112	0.322	1.022		0.084	0.294	1.204		0.28	
33	As	[Ar] $4s^2 3d^{10} 4p^3$	0.014	0.098	0.308	0.952		0.07	0.294	1.078		0.266	
34	Se	[Ar] $4s^2 3d^{10} 4p^4$	0.014	0.098	0.294	0.896		0.07	0.28	0.98		0.252	
35	Br	[Ar] $4s^2 3d^{10} 4p^5$	0.014	0.098	0.294	0.84		0.07	0.266	0.91		0.238	
36	Kr	[Ar] $4s^2 3d^{10} 4p^6$	0.014	0.098	0.28	0.798		0.07	0.252	0.854		0.224	
37	Rb	[Kr] $5s^1$	0.014	0.098	0.266	0.756	2.52	0.07	0.252	0.784		0.224	
38	Sr	[Kr] $5s^2$	0.014	0.084	0.266	0.714	2.17	0.07	0.238	0.728		0.21	
39	Y	[Kr] $5s^2 4d^1$	0.014	0.084	0.252	0.672	2.002	0.056	0.224	0.686		0.196	1.008
40	Zr	[Kr] $5s^2 4d^2$	0.014	0.084	0.252	0.644	1.89	0.056	0.224	0.644		0.196	0.896
41	Nb	[Kr] $5s^1 4d^4$	0.014	0.084	0.238	0.616	1.876	0.056	0.21	0.616		0.182	0.84
42	Mo	[Kr] $5s^1 4d^5$	0.014	0.084	0.238	0.602	1.806	0.056	0.21	0.588		0.182	0.77
43	Tc	[Kr] $5s^1 4d^5$	0.014	0.084	0.224	0.574	1.666	0.056	0.196	0.56		0.168	0.714
44	Ru	[Kr] $5s^2 4d^5$	0.014	0.084	0.224	0.56	1.694	0.056	0.196	0.532		0.168	0.686
45	Rh	[Kr] $5s^1 4d^7$	0.014	0.07	0.21	0.532	1.652	0.056	0.182	0.518		0.168	0.644
46	Pd	[Kr] $5s^1 4d^8$	0.014	0.07	0.21	0.518	1.722	0.056	0.182	0.49		0.154	0.616
47	Ag	[Kr] $4d^{10}$	0.014	0.07	0.21	0.504	1.582	0.056	0.182	0.476		0.154	0.588
48	Cd	[Kr] $5s^2 4d^{10}$	0.014	0.07	0.196	0.49	1.47	0.042	0.168	0.462		0.154	0.546
49	In	[Kr] $5s^2 4d^{10} 5p^1$	0.014	0.07	0.196	0.476	1.358	0.042	0.168	0.448	1.47	0.14	0.518
50	Sn	[Kr] $5s^2 4d^{10} 5p^2$	0.014	0.07	0.196	0.462	1.274	0.042	0.196	0.434	1.344	0.14	0.504

Table 4. The choices of the cutoff radius r_c (Å) for each states of z=51 to z=75 atoms.

z	atom	electronic state	1s	2s	3s	4s	5s	6s	2p	3p	4p	5p	3d	4d	5d	4f
51	Sb	[Kr] $5s^2 4d^{10} 5p^3$	0.014	0.07	0.182	0.448	1.204		0.042	0.154	0.42	1.232	0.14	0.476		
52	Te	[Kr] $5s^2 4d^{10} 5p^4$	0.014	0.07	0.182	0.434	1.134		0.042	0.154	0.406	1.148	0.14	0.448		
53	I	[Kr] $5s^2 4d^{10} 5p^5$	0.014	0.07	0.182	0.42	1.078		0.042	0.154	0.392	1.078	0.126	0.434		
54	Xe	[Kr] $5s^2 4d^{10} 5p^6$	0.014	0.07	0.182	0.42	1.036		0.042	0.154	0.378	1.022	0.126	0.42		
55	Cs	[Xe] $6s^1$	0.014	0.07	0.168	0.406	0.98	2.912	0.042	0.14	0.364	0.952	0.126	0.406		
56	Ba	[Xe] $6s^2$	0.014	0.07	0.168	0.392	0.938	2.59	0.042	0.14	0.35	0.896	0.126	0.392		
57	La	[Xe] $5d^1 6s^2$	0.014	0.056	0.168	0.378	0.896	2.422	0.042	0.14	0.336	0.854	0.112	0.378	1.246	
58	Ce	[Xe] $4f^1 5d^1 6s^2$	0.014	0.056	0.168	0.378	0.882	2.408	0.042	0.14	0.336	0.826	0.112	0.364	1.246	0.406
59	Pr	[Xe] $4f^3 6s^2$	0.014	0.056	0.168	0.364	0.854	2.282	0.042	0.126	0.322	0.798	0.112	0.364		0.378
60	Nd	[Xe] $4f^4 6s^2$	0.014	0.056	0.154	0.364	0.854	2.436	0.042	0.126	0.322	0.798	0.112	0.35		0.378
61	Pm	[Xe] $4f^5 6s^2$	0.014	0.056	0.154	0.35	0.84	2.422	0.042	0.126	0.308	0.77	0.112	0.336		0.35
62	Sm	[Xe] $4f^6 6s^2$	0.014	0.056	0.154	0.35	0.826	2.394	0.028	0.126	0.294	0.756	0.112	0.336		0.35
63	Eu	[Xe] $4f^7 6s^2$	0.014	0.056	0.154	0.336	0.812	2.38	0.028	0.126	0.294	0.742	0.098	0.322		0.336
64	Gd	[Xe] $4f^7 5d^1 6s^2$	0.014	0.056	0.154	0.336	0.798	2.254	0.028	0.112	0.28	0.714	0.098	0.308	1.106	0.322
65	Tb	[Xe] $4f^9 6s^2$	0.014	0.056	0.14	0.336	0.784	2.324	0.028	0.112	0.28	0.7	0.098	0.308		0.308
66	Dy	[Xe] $4f^{10} 6s^2$	0.014	0.056	0.14	0.322	0.784	2.31	0.028	0.112	0.28	0.686	0.098	0.294		0.308
67	Ho	[Xe] $4f^{11} 6s^2$	0.014	0.056	0.14	0.322	0.784	2.352	0.028	0.112	0.266	0.294	0.098	0.294		0.294
68	Er	[Xe] $4f^{12} 6s^2$	0.014	0.056	0.14	0.308	0.742	1.96	0.028	0.112	0.266	0.644	0.098	0.294		0.28
69	Tm	[Xe] $4f^{13} 6s^2$	0.014	0.056	0.14	0.308	0.756	2.212	0.028	0.112	0.252	0.658	0.098	0.28		0.294
70	Yb	[Xe] $4f^{14} 6s^2$	0.014	0.056	0.14	0.308	0.714	1.862	0.028	0.098	0.252	0.616	0.084	0.28		0.266
71	Lu	[Xe] $4f^{14} 6s^2 5d^1$	0.014	0.056	0.14	0.294	0.728	2.142	0.028	0.098	0.252	0.616	0.084	0.266	1.036	0.266
72	Hf	[Xe] $4f^{14} 6s^2 5d^2$	0.014	0.056	0.14	0.294	0.7	2.044	0.028	0.098	0.238	0.602	0.084	0.266	0.952	0.252
73	Ta	[Xe] $4f^{14} 6s^2 5d^3$	0.014	0.056	0.126	0.294	0.686	1.96	0.028	0.098	0.238	0.574	0.084	0.266	0.896	0.252
74	W	[Xe] $4f^{14} 6s^2 5d^4$	0.014	0.056	0.126	0.294	0.672	1.904	0.028	0.098	0.224	0.56	0.084	0.252	0.84	0.238
75	Re	[Xe] $4f^{14} 6s^2 5d^5$	0.014	0.056	0.126	0.28	0.658	1.848	0.028	0.098	0.224	0.546	0.084	0.252	0.798	0.238

Table 5. The choices of the cutoff radius r_c (Å) for each states of z=76 to z=102 atoms.

Z	atom	electronic state	1s	2s	3s	4s	5s	6s	7s	2p	3p	4p	5p	6p	3d	4d	5d	6d	4f	5f
76	Os	[Xe] $4f^{14}6s^2 5d^6$	0.014	0.056	0.126	0.28	0.644	1.806		0.028	0.098	0.224	0.532		0.084	0.252	0.77		0.224	
77	Ir	[Xe] $4f^{14}6s^2 5d^7$	0.014	0.056	0.126	0.28	0.63	1.764		0.028	0.084	0.21	0.518		0.084	0.238	0.728		0.224	
78	Pt	[Xe] $4f^{14}6s^1 5d^9$	0.014	0.042	0.126	0.266	0.616	1.694		0.028	0.084	0.21	0.49		0.084	0.238	0.7		0.21	
79	Au	[Xe] $4f^{14}6s^1 5d^{10}$	0.014	0.042	0.126	0.266	0.602	1.806		0.028	0.084	0.21	0.49		0.084	0.238	0.686		0.21	
80	Hg	[Xe] $4f^{14}6s^2 5d^{10}$	0.014	0.042	0.126	0.266	0.588	1.68		0.028	0.084	0.21	0.476		0.07	0.224	0.658		0.21	
81	Tl	[Xe] $4f^{14}6s^2 5d^{10}6p^1$	0.014	0.042	0.126	0.266	0.574	1.582		0.028	0.084	0.196	0.462	1.316	0.07	0.224	0.63		0.196	
82	Pb	[Xe] $4f^{14}6s^2 5d^{10}6p^2$	0.014	0.042	0.112	0.252	0.56	1.498		0.028	0.084	0.196	0.448	1.232	0.07	0.224	0.602		0.196	
83	Bi	[Xe] $4f^{14}6s^2 5d^{10}6p^3$	0.014	0.042	0.112	0.252	0.546	1.442		0.028	0.084	0.196	0.434	1.162	0.07	0.21	0.588		0.196	
84	Po	[Xe] $4f^{14}6s^2 5d^{10}6p^4$	0.014	0.042	0.112	0.252	0.546	1.372		0.028	0.084	0.196	0.42	1.106	0.07	0.21	0.56		0.182	
85	At	[Xe] $4f^{14}6s^2 5d^{10}6p^5$	0.014	0.042	0.112	0.252	0.532	1.33		0.028	0.084	0.182	0.42	1.05	0.07	0.21	0.546		0.182	
86	Rn	[Xe] $4f^{14}6s^2 5d^{10}6p^6$	0.014	0.042	0.112	0.238	0.518	1.288		0.028	0.084	0.182	0.406	1.008	0.07	0.21	0.532		0.182	
87	Fr	[Rn] $7s^1$	0.014	0.042	0.112	0.238	0.504	1.204	3.304	0.028	0.07	0.182	0.392	0.952	0.07	0.196	0.518		0.182	
88	Ra	[Rn] $7s^2$	0.014	0.042	0.112	0.238	0.504	1.148	3.01	0.028	0.07	0.182	0.392	0.91	0.07	0.196	0.504		0.168	
89	Ac	[Rn] $7s^2 6d^1$	0.014	0.042	0.112	0.238	0.49	1.106	2.828	0.028	0.07	0.168	0.378	0.868	0.07	0.196	0.49	1.47	0.168	
90	Th	[Rn] $7s^2 6d^2$	0.014	0.042	0.112	0.238	0.476	1.064	2.688	0.014	0.07	0.168	0.364	0.84	0.07	0.196	0.476	1.358	0.168	
91	Pa	[Rn] $7s^2 5f^2 6d^1$	0.014	0.042	0.112	0.224	0.476	1.05	2.73	0.014	0.07	0.168	0.364	0.812	0.07	0.196	0.462	1.372	0.168	0.63
92	U	[Rn] $7s^2 5f^3 6d^1$	0.014	0.042	0.112	0.224	0.462	1.036	2.702	0.014	0.07	0.168	0.35	0.798	0.07	0.182	0.448	1.344	0.168	0.602
93	Np	[Rn] $7s^2 5f^4 6d^1$	0.014	0.042	0.112	0.224	0.462	1.022	2.674	0.014	0.07	0.168	0.35	0.77	0.07	0.182	0.434	1.316	0.154	0.574
94	Pu	[Rn] $7s^2 5f^6$	0.014	0.042	0.098	0.224	0.448	1.022	2.758	0.014	0.07	0.154	0.336	0.77	0.07	0.182	0.434		0.154	0.56
95	Am	[Rn] $7s^2 5f^7$	0.014	0.042	0.098	0.224	0.448	1.008	2.744	0.014	0.07	0.154	0.336	0.742	0.056	0.182	0.42		0.154	0.546
96	Cm	[Rn] $7s^2 5f^7 6d^1$	0.014	0.042	0.098	0.21	0.434	0.98	2.618	0.014	0.07	0.154	0.322	0.728	0.056	0.182	0.42	1.246	0.154	0.518
97	Bk	[Rn] $7s^2 5f^9$	0.014	0.042	0.098	0.21	0.434	0.98	2.716	0.014	0.07	0.154	0.322	0.714	0.056	0.168	0.406		0.154	0.504
98	Cf	[Rn] $7s^2 5f^{10}$	0.014	0.042	0.098	0.21	0.42	0.966	2.702	0.014	0.07	0.154	0.322	0.7	0.056	0.168	0.392		0.14	0.49
99	Es	[Rn] $7s^2 5f^{11}$	0.014	0.042	0.098	0.21	0.42	0.952	2.688	0.014	0.07	0.154	0.308	0.686	0.056	0.168	0.392		0.14	0.476
100	Fm	[Rn] $7s^2 5f^{12}$	0.014	0.042	0.098	0.21	0.42	0.952	2.688	0.014	0.056	0.14	0.308	0.672	0.056	0.168	0.378		0.14	0.462
101	Md	[Rn] $7s^2 5f^{13}$	0.014	0.042	0.098	0.21	0.406	0.938	2.674	0.014	0.056	0.14	0.294	0.658	0.056	0.168	0.378		0.14	0.448
102	No	[Rn] $7s^2 5f^{14}$	0.014	0.042	0.098	0.21	0.406	0.924	2.66	0.014	0.056	0.14	0.294	0.644	0.056	0.168	0.364		0.14	0.448

An intermediate pseudo potential V_1^{ps}:

$$V_1^{ps}(r) = \left[1 - f(\frac{r}{r_c})\right] V_l(r) + c_1 f(\frac{r}{r_c}) \tag{3.24}$$

was first formed with the corresponding the radial Schrödinger equation:

$$-\frac{1}{2}\frac{d^2 w_1(r)}{d^2 r} + V_1^{ps}(r) w_1(r) = \varepsilon_1 w_1(r) \,, \tag{3.25}$$

where introducing a cut-off function $f(x)$ which approaches 0 as $x \to \infty$, approaches 1 at least as fast as x^3 as $x \to 0$, and cuts off for $x \approx 1$. A convenient choice was:

$$f(x) = \exp(-x^4)$$

The constant c_1 was adjusted so that the **nodeless** bound solution w_1 of the radial Schrödinger equation (3.25) with $V_1^{ps}(r)$ has energy ε_1 equal to the original eigenvalue of the valence state ε_{nl}. By using the secant method, we can do loop to get c_1.

Property (1) of the subsection 2.4 is now automatically satisfied. Further, property (2) of subsection 2.4 is satisfied within a multiplicative constant γ_1 for normalized function w_1:

$$\gamma_1 w_1(r) \underset{r>r_c}{\longrightarrow} p_{nl}(r) \tag{3.26}$$

Since both satisfy the same differential equation for $r > r_c$ and with the same homogeneous boundary condition. Here we choose

$$\gamma_1 = \frac{P_{nl}(a)}{w_1(a)},$$

where a is about the middle point($\approx 5\mathring{A}$) of the domain ($0\sim10\mathring{A}$).

To satisfy properties (2)-(4) of the subsection 2.4, the intermediate pseudo wave function w_1 is modified to

$$w_2(r) = \gamma_1 \left[w_1(r) + \delta_l g_l(\frac{r}{r_c}) \right] \quad , \tag{3.27}$$

where another cut off function $g_l(x)$ was introduced, which cuts off to zero for x>1, and behaves as x^{l+1} at small x. The convenient choice is written as:

$$g_l(x) = x^{l+1} \exp(-x^4) .$$

The asymptotic behavior of $f(x)$ and $g_l(x)$ guarantees the potential to be finite at the origin. We choose δ_l that is the smaller solution of the quadratic equation of the normalization condition:

$$\int_0^\infty \gamma_1^2 \left[w_1(r) + \delta_l g_l(\frac{r}{r_c}) \right]^2 dr = 1 \tag{3.28}$$

Expanding the equation (3.28), we have:

$$\left[\gamma_1^2 \int_0^\infty g_l^2(\frac{r}{r_c}) dr \right] \delta_l^2 + \left[2\gamma_1^2 \int_0^\infty w_1(r) g_l(\frac{r}{r_c}) dr \right] \delta_l + \left[\gamma_1^2 \int_0^\infty w_1^2(r) dr - 1 \right] = 0 \tag{3.29}$$

Solving the quadratic equation (3.29), we can get smaller solution δ_l.

Notice that this condition (3.28) guarantees that the norm conservation condition because for w_2 with P_{nl} since both function coincide for $r > r_c$. The pseudopotential $V_{ps}(r)$ producing the **nodeless** eigenfunction $w_2(r)$ at eigenvalue ε_{nl} is now found by inverting the radial Schrödinger equation:

$$V_{ps}(r) = \left[\varepsilon_{nl} w_2(r) + \frac{1}{2} \frac{d^2 w_2(r)}{d^2 r} \right] \frac{1}{w_2(r)} \tag{3.30}$$

To form the final bare-ion pseudopotential, the valence pseudo charge density is found by using the wave functions w_2 in the same configuration as the original atom calculation. The Hartree and exchange-correlation potentials due to this density are then calculated and subtracted from each V_{ps}. Analytical expressions containing a few parameters can subsequently be fitted to the numerical potential functions.

A Appendix

A.1 The main program of No atom

```
module toole
implicit none
real(kind=8)::z1=102.0    !atomic number of NO
real::a0=0.529 !Bohr radius
real,parameter::pi=3.14159265
integer,parameter::g=400
real,parameter::h=10.0/real(g) !切割間距
real,parameter::h1=10.0 !範圍 0-h1
real(kind=8)
a(g,g),v_l(g),v1l_ps(g),v2l_ps(g),e_nl,r_c,c_l,r1,delta_l,w10(0:g),w20(0:g),pnl(0:g)

real(kind=8)
v(g),r_s(0:g),v_xc(0:g),v_ee(0:g),v_eff(g),e_xc(0:g),e_x(0:g),e_c(0:g),electron,electron1
,ev0(7),ev1(6),ev2(4),ev3(2)

real(kind=8)
k(0:g),p_10(0:g),p_20(0:g),p_30(0:g),p_40(0:g),p_21(0:g),p_31(0:g),p_32(0:g),p_41(0:g)
,p_50(0:g),p_42(0:g),p_51(0:g),p_60(0:g),p_52(0:g)

real(kind=8) p_43(0:g),p_61(0:g),p_70(0:g),p_62(0:g),p_53(0:g)

real(kind=8)
f(0:g,0:g),f1(0:g),diffr_s(0:g),f2(g,0:g),f2_1(0:g),f3(0:g),f4(0:g),n(0:g),n_out(0:g),dex4(0:g)
,dex5(0:g)

real(kind=8)
dex_10(g),dex_20(g),dex_30(g),dex_40(g),dex_21(g),dex_31(g),dex_32(g),dex_41(g)
,dex_50(g),dex_42(g),dex_51(g),dex_60(g),dex_52(g)

real(kind=8) dex_43(g),dex_61(g),dex_70(g),dex_62(g),dex_53(g)
integer key,ierr
double precision w(g-1),e(g-1),z(g-1,g-1)
real::l
```

```
real(kind=8) kinetic_energy,E_xc_functional,external_energy,coulomb_energy,total_energy
end module

program main
use toole
implicit none
real(kind=8) norm,norm1
integer i,count,error
real*8,external::normal_P
logical alive

call gp()

inquire(file="DENSITY MD.txt",exist=alive)
if (.not. alive) then
write(*,*) trim("DENSITY MD.txt"),"doesn't exist."
end if
open(10,file="DENSITY MD.txt")
do while(.true.)
read(10,*,iostat=error)n
if (error/=0) exit
end do
close(10)



do while (.true.)
norm=0.0
norm1=0.0
call sub11()
call sub12()
    if (abs(electron-z1)<1e-3) then
        n=n
     else
        n=(z1/electron)*n
    end if
```

```
call radius()    !the radius r_s of a unit charge
call ee()          !the exchange energy e_x
call ce()          !the correlation energy    e_c
call ece()         !the exchange-correlation energy per unit density e_xc
call sub0()      !diff. of r_s
call ecp()       !The exchange-correlation potential v_xc
call sub10()    !計算 electron-electron coulomb potential 中的項
call sub10_1()
call eecp()      !the electron-electron coulomb potential v_ee
call ep()        !external potential     v
call effp()      !effective potential v_eff

l=0.0
call computing_ev_evt() !計算 eigenvalue & eigenvector
p_10(1:g-1)=z(:,1)
p_20(1:g-1)=z(:,2)
p_30(1:g-1)=z(:,3)
p_40(1:g-1)=z(:,4)
p_50(1:g-1)=z(:,5)
p_60(1:g-1)=z(:,6)
p_70(1:g-1)=z(:,7)
p_10=sqrt(1.0/normal_P(p_10))*p_10
p_20=sqrt(1.0/normal_P(p_20))*p_20
p_30=sqrt(1.0/normal_P(p_30))*p_30
p_40=sqrt(1.0/normal_P(p_40))*p_40
p_50=sqrt(1.0/normal_P(p_50))*p_50
p_60=sqrt(1.0/normal_P(p_60))*p_60
p_70=sqrt(1.0/normal_P(p_70))*p_70
ev0(1)=w(1)
ev0(2)=w(2)
ev0(3)=w(3)
ev0(4)=w(4)
ev0(5)=w(5)
ev0(6)=w(6)
ev0(7)=w(7)

l=1.0
call computing_ev_evt() !計算 eigenvalue & eigenvector
p_21(1:g-1)=z(:,1)
```

```
p_31(1:g-1)=z(:,2)
p_41(1:g-1)=z(:,3)
p_51(1:g-1)=z(:,4)
p_61(1:g-1)=z(:,5)
p_21=sqrt(1.0/normal_P(p_21))*p_21
p_31=sqrt(1.0/normal_P(p_31))*p_31
p_41=sqrt(1.0/normal_P(p_41))*p_41
p_51=sqrt(1.0/normal_P(p_51))*p_51
p_61=sqrt(1.0/normal_P(p_61))*p_61
ev1(1)=w(1)
ev1(2)=w(2)
ev1(3)=w(3)
ev1(4)=w(4)
ev1(5)=w(5)

l=2.0
call computing_ev_evt() !計算 eigenvalue & eigenvector
p_32(1:g-1)=z(:,1)
p_42(1:g-1)=z(:,2)
p_52(1:g-1)=z(:,3)
p_62(1:g-1)=z(:,4)
p_32=sqrt(1.0/normal_P(p_32))*p_32
p_42=sqrt(1.0/normal_P(p_42))*p_42
p_52=sqrt(1.0/normal_P(p_52))*p_52
p_62=sqrt(1.0/normal_P(p_62))*p_62
ev2(1)=w(1)
ev2(2)=w(2)
ev2(3)=w(3)
ev2(4)=w(4)

l=3.0
call computing_ev_evt() !計算 eigenvalue & eigenvector
p_43(1:g-1)=z(:,1)
p_53(1:g-1)=z(:,2)
p_43=sqrt(1.0/normal_P(p_43))*p_43
p_53=sqrt(1.0/normal_P(p_53))*p_53
ev3(1)=w(1)
ev3(2)=w(2)
```

```
   call density_out() !density n_out

      do i=0,g
          norm=norm+(n(i)-n_out(i))**2.0
          norm1=norm1+n(i)**2.0
      end do
   if(sqrt(norm/norm1)<1e-4) then

      exit
   else
      do i=0,g
        n(i)=0.5*n_out(i)+0.5*n(i)
      end do
   end if

end do

open(unit=10,file='energy.txt')
call E_G() !Total energy E_G[n]

write(10,*) "動能=",kinetic_energy
write(10,*) "coulomb_energy=",coulomb_energy
write(10,*) "exchange-correlation fuctional E_xc=",E_xc_functional
write(10,*) "external_energy=",external_energy
write(10,*) "總能=",total_energy
write(10,*) "誤差=",(abs(-total_energy-32790)/32790)*100.0,"%"

write(*,*) "動能=",kinetic_energy
write(*,*) "coulomb_energy=",coulomb_energy
write(*,*) "exchange-correlation fuctional E_xc=",E_xc_functional
write(*,*) "external_energy=",external_energy
write(*,*) "總能=",total_energy



open(unit=10,file='DENSITY NO.txt')
write(10,*) (n_out(i),i=0,g)
```

```
l=3.0 !改
pnl=p_53 !改
e_nl=ev3(2) !改
call find_c_l01()
call sub15()
call sub16()
call computing_ev_evt_ps()
w10(1:g-1)=z(:,1)
w10=sqrt(1.0/normal_P(w10))*w10
r1=p_53(200)/w10(200) !改
call find_delta_l()
call sub_w2_0()
call pseudo()
open(unit=10,file='w20 p_53.txt') !改
 do i=0,g
   write(10,*)w20(i)
 end do
open(unit=10,file='v21_ps   p_53.txt') !改
 do i=1,g-1
   write(10,*)v21_ps(i)
 end do




end



subroutine gp() !grid point
use toole
implicit none
integer i
 do i=0,g
      k(i)=(real(i)/real(g))*h1
 end do
end
```

```
subroutine radius() !the radius r_s of a unit charge
use toole
implicit none
integer i
  do i=0,g-1
      r_s(i)=(3.0/(4.0*pi*n(i)))**(1.0/3.0)
  end do
end

subroutine ee() !the exchange energy
use toole
implicit none
integer i
  do i=0,g-1
      e_x(i)=-0.458/r_s(i)
  end do
end

subroutine ce() !the correlation energy
use toole
implicit none
integer i
do i=0,g-1
    if (r_s(i)>=1.0) then
         e_c(i)=-0.1423/(1.0+1.0529*sqrt(r_s(i))+0.3334*r_s(i))
    else if (r_s(i)<1.0) then
         e_c(i)=0.0311*log(r_s(i))+(-0.0480)+0.002*r_s(i)*log(r_s(i))+(-0.0116)*r_s(i)
    end if
end do
end

subroutine ece() !the exchange-correlation energy per unit density
use toole
implicit none
integer i
  do i=0,g-1
      e_xc(i)=e_x(i)+e_c(i)
```

```
  end do
end

subroutine sub0() !diff. of r_s
use toole
implicit none
integer i
   do i=0,g-1
        diffr_s(i)=-4.0*pi*3.0**(-2.0/3.0)*(4.0*pi*n(i))**(-4.0/3.0)
   end do
end

subroutine ecp() !The exchange-correlation potential v_xc
use toole
implicit none
integer i
   do i=0,g-1
      if (r_s(i)>=1.0) then

v_xc(i)=e_xc(i)+n(i)*(0.458*r_s(i)**(-2.0)*diffr_s(i)+0.1423*(1.0+1.0529*r_s(i)**(0.5)+0.3
334*r_s(i))**(-2.0)*(0.5*1.0529*r_s(i)**(-0.5)*diffr_s(i)+0.3334*diffr_s(i)))

        else if (r_s(i)<1.0) then

v_xc(i)=e_xc(i)+n(i)*(0.458*r_s(i)**(-2.0)*diffr_s(i)+0.0311*r_s(i)**(-1.0)*diffr_s(i)+0.002
*diffr_s(i)*log(r_s(i))+0.002*diffr_s(i)-0.0116*diffr_s(i))
      end if
   end do
end



subroutine ep() !external potential v
use toole
implicit none
integer i
  do i=1,g
     v(i)=-z1/k(i)
```

```
  end do
end

subroutine eecp() !The electron-electron coulomb potential V_ee
use toole                    !v_ee(r)= ∫(0.0,10.0)n(r1)*(r1**2.0)*1.0/max(r,r1)
implicit none                !f2 from sub10()
integer i
real,external::trape_integral4,trape_integral5
  do i=0,g
    if(i==0) then
      v_ee(i)=4.0*pi*trape_integral4(f2_1,h,0)
     else
      v_ee(i)=4.0*pi*trape_integral5(f2,h,i)
    end if
  end do
end


subroutine effp() !effective potential v_eff
use toole
implicit none
integer i
  do i=1,g-1
     v_eff(i)=v(i)+v_xc(i)+v_ee(i)
  end do
end

real function trape_integral3(datas2,width,j) !simpson 積分
use toole
implicit none
real(kind=8) datas2(0:g,0:g)
real width
real sum
integer i,j
sum=(datas2(0,j)+datas2(g,j))
  do i=1,g-1
    if (mod(i,2)==0) then
      sum=sum+4.0*datas2(i,j)
    else
```

```
        sum=sum+2.0*datas2(i,j)
      end if
  end do
trape_integral3=sum*width/3.0
return
end

real function trape_integral4(datas,width,j) !simpson 積分
use toole
implicit none
real(kind=8) datas(0:g)
real width
real sum
integer i,j
sum=(datas(j)+datas(g))
  do i=j+1,g-1
      if (mod(i,2)==0) then
        sum=sum+4.0*datas(i)
      else
        sum=sum+2.0*datas(i)
      end if
  end do
trape_integral4=sum*width/3.0
return
end

real function trape_integral41(datas,width,j) !simpson 積分
use toole
implicit none
real(kind=8) datas(g)
real width
real sum
integer i,j
sum=(datas(j)+datas(g-1))
  do i=j+1,g-2
      if (mod(i,2)==0) then
        sum=sum+4.0*datas(i)
      else
        sum=sum+2.0*datas(i)
```

```
      end if
  end do
trape_integral41=sum*width/3.0
return
end

real function trape_integral5(datas2,width,i) !simpson 積分
use toole
implicit none
real(kind=8) datas2(g,0:g)
real width
real sum
integer i,j
sum=(datas2(i,0)+datas2(i,g))
  do j=1,g-1
      if (mod(j,2)==0) then
          sum=sum+4.0*datas2(i,j)
      else
          sum=sum+2.0*datas2(i,j)
      end if
  end do
trape_integral5=sum*width/3.0
return
end


subroutine computing_ev_evt()
use toole                              !計算 eigenvalue and eigenvector
implicit none
integer:: i,j
do j=1,g-1      !離散化之後產生的矩陣
  do i=1,g-1
      if (i==j) then
           a(i,j)=(1.0/(h**2.0)+l*(l+1.0)/(2.0*k(i)**2.0)+v_eff(i))
        else if (abs(i-j)==1) then
           a(i,j)=(-1.0/(2.0*h**2.0))
        else
           a(i,j)=0.0
      end if
```

```
    end do
end do

  do i=1,g-1
        w(i)=a(i,i)
  end do

  do i=2,g-1
        e(i)=(-1.0/(2.0*h**2.0))
  end do
call rst(g-1,g-1,w,e,1,z,ierr)

end subroutine



subroutine sub3() !計算動能的項
use toole
implicit none
integer i
l=0.0
do i=1,g-1
        dex_10(i)=(-1.0/2.0)*p_10(i)*(p_10(i+1)-2.0*p_10(i)+p_10(i-1))/(h**2.0)&
                        &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_10(i)**2.0
end do

    do i=1,g-1
      dex_20(i)=(-1.0/2.0)*p_20(i)*(p_20(i+1)-2.0*p_20(i)+p_20(i-1))/(h**2.0)&
                      &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_20(i)**2.0

    end do

    do i=1,g-1
      dex_30(i)=(-1.0/2.0)*p_30(i)*(p_30(i+1)-2.0*p_30(i)+p_30(i-1))/(h**2.0)&
                      &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_30(i)**2.0

    end do

    do i=1,g-1
```

```
   dex_40(i)=(-1.0/2.0)*p_40(i)*(p_40(i+1)-2.0*p_40(i)+p_40(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_40(i)**2.0

  end do

  do i=1,g-1
   dex_50(i)=(-1.0/2.0)*p_50(i)*(p_50(i+1)-2.0*p_50(i)+p_50(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_50(i)**2.0
  end do

  do i=1,g-1
   dex_60(i)=(-1.0/2.0)*p_60(i)*(p_60(i+1)-2.0*p_60(i)+p_60(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_60(i)**2.0
  end do

  do i=1,g-1
   dex_70(i)=(-1.0/2.0)*p_70(i)*(p_70(i+1)-2.0*p_70(i)+p_70(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_70(i)**2.0
  end do

 l=1.0
  do i=1,g-1
   dex_21(i)=(-1.0/2.0)*p_21(i)*(p_21(i+1)-2.0*p_21(i)+p_21(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_21(i)**2.0

  end do

  do i=1,g-1
   dex_31(i)=(-1.0/2.0)*p_31(i)*(p_31(i+1)-2.0*p_31(i)+p_31(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_31(i)**2.0

  end do

  do i=1,g-1
   dex_41(i)=(-1.0/2.0)*p_41(i)*(p_41(i+1)-2.0*p_41(i)+p_41(i-1))/(h**2.0)&
           &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_41(i)**2.0

  end do
```

```
  do i=1,g-1
    dex_51(i)=(-1.0/2.0)*p_51(i)*(p_51(i+1)-2.0*p_51(i)+p_51(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_51(i)**2.0

  end do

  do i=1,g-1
    dex_61(i)=(-1.0/2.0)*p_61(i)*(p_61(i+1)-2.0*p_61(i)+p_61(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_61(i)**2.0

  end do

 l=2.0
 do i=1,g-1
    dex_32(i)=(-1.0/2.0)*p_32(i)*(p_32(i+1)-2.0*p_32(i)+p_32(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_32(i)**2.0

  end do

  do i=1,g-1
    dex_42(i)=(-1.0/2.0)*p_42(i)*(p_42(i+1)-2.0*p_42(i)+p_42(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_42(i)**2.0

  end do

  do i=1,g-1
    dex_52(i)=(-1.0/2.0)*p_52(i)*(p_52(i+1)-2.0*p_52(i)+p_52(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_52(i)**2.0

  end do

  do i=1,g-1
    dex_62(i)=(-1.0/2.0)*p_62(i)*(p_62(i+1)-2.0*p_62(i)+p_62(i-1))/(h**2.0)&
                &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_62(i)**2.0

  end do

 l=3.0
 do i=1,g-1
```

```
    dex_43(i)=(-1.0/2.0)*p_43(i)*(p_43(i+1)-2.0*p_43(i)+p_43(i-1))/(h**2.0)&
            &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_43(i)**2.0

   end do

   do i=1,g-1
    dex_53(i)=(-1.0/2.0)*p_53(i)*(p_53(i+1)-2.0*p_53(i)+p_53(i-1))/(h**2.0)&
            &+(l*(l+1.0)/(2.0*k(i)**2.0))*p_53(i)**2.0

   end do
end

subroutine KE() !動能 T
use toole
implicit none
real,external::trape_integral41
kinetic_energy=2.0*trape_integral41(dex_10,h,1)+2.0*trape_integral41(dex_20,h,1)+6.0*trap
             e_integral41(dex_21,h,1)+2.0*trape_integral41(dex_30,h,1)&
             &+6.0*trape_integral41(dex_31,h,1)+2.0*trape_integral41(dex_40,h,1)+10.0
             *trape_integral41(dex_32,h,1)+6.0*trape_integral41(dex_41,h,1)&
             &+2.0*trape_integral41(dex_50,h,1)+10.0*trape_integral41(dex_42,h,1)+6.0
             *trape_integral41(dex_51,h,1)+2.0*trape_integral41(dex_60,h,1)&
             &+10.0*trape_integral41(dex_52,h,1)+14.0*trape_integral41(dex_43,h,1)+6.
             0*trape_integral41(dex_61,h,1)+2.0*trape_integral41(dex_70,h,1)&
             &+0.0*trape_integral41(dex_62,h,1)+14.0*trape_integral41(dex_53,h,1)
                                     !                              !
end

subroutine sub4() !計算 exchange-correlation fuctional E_xc 中使用
use toole
implicit none
integer i
  do i=0,g
   if(i==g) then
      dex4(i)=0.0
    else
      dex4(i)=e_xc(i)*n_out(i)*(k(i)**2.0)!
   end if
  end do
```

```
end

subroutine Excfunctional() !exchange-correlation fuctional E_xc
use toole
implicit none
real,external::trape_integral4
E_xc_functional=4.0*pi*trape_integral4(dex4,h,0) !
End

subroutine sub5() !計算 external energy ∫v(r)n(r)dr 使用
use toole
implicit none
integer i
 do i=0,g
     dex5(i)=z1*k(i)*n_out(i) !
 end do
end

subroutine sub6() !計算總能中的 ∫v(r)n(r)dr
use toole
implicit none
real,external::trape_integral4
external_energy=-4.0*pi*trape_integral4(dex5,h,0)
end


subroutine sub7() !計算 coulomb energy 使用
use toole
implicit none
integer i,j
do j=0,g
 do i=0,g
   if(i==0 .and. j==0) then
    f(i,j)=0.0
    else
    f(i,j)=(k(i)**2.0)*(k(j)**2.0)*(1.0/max(k(i),k(j)))*n_out(i)*n_out(j)
   end if
 end do
end do
```

```
end

subroutine sub8()              !計算 coulomb energy 使用
use toole
implicit none
integer j
real,external::trape_integral3
 do j=0,g
      f1(j)=trape_integral3(f,h,j)
 end do
end


subroutine sub9() !coulomb energy (1/2)*∫∫(Vee*n)dr
use toole
implicit none
real,external::trape_integral4
coulomb_energy=0.5*(4.0*pi)**2.0*trape_integral4(f1,h,0)
end

subroutine sub10()
use toole
implicit none
integer i,j
 do j=1,g
   do i=1,g
      f2(i,j)=k(j)**2.0*n(j)*(1.0/max(k(i),k(j)))
   end do
 end do
end

subroutine sub10_1()
use toole
implicit none
integer i
 do i=0,g
    f2_1(i)=k(i)*n(i)
 end do
```

```
end

subroutine sub11()
use toole
implicit none
integer i
   do i=0,g
     f3(i)=k(i)**2.0*n(i)
   end do
end

subroutine sub12()
use toole
implicit none
real,external::trape_integral4
electron=4.0*pi*trape_integral4(f3,h,0)
end


real*8 function normal_P(p_nl)
use toole
implicit none
real(kind=8) p_nl(0:g),temp(0:g)
integer i
real,external::trape_integral4
  do i=0,g
    temp(i)=P_nl(i)**2.0
  end do
normal_P=trape_integral4(temp,h,0)
end

subroutine density_out() !density n_out
use toole
implicit none
integer i



        p_10(0)=0.0
```

```
        p_20(0)=0.0
        p_30(0)=0.0
        p_21(0)=0.0
        p_31(0)=0.0
        p_40(0)=0.0
        p_32(0)=0.0
        p_41(0)=0.0
        p_50(0)=0.0
        p_42(0)=0.0
        p_51(0)=0.0
        p_60(0)=0.0
        p_52(0)=0.0
        p_43(0)=0.0
        p_61(0)=0.0
        p_70(0)=0.0
        p_53(0)=0.0

        p_10(g)=0.0
        p_20(g)=0.0
        p_30(g)=0.0
        p_21(g)=0.0
        p_31(g)=0.0
        p_40(g)=0.0
        p_32(g)=0.0
        p_41(g)=0.0
        p_50(g)=0.0
        p_42(g)=0.0
        p_51(g)=0.0
        p_60(g)=0.0
        p_52(g)=0.0
        p_43(g)=0.0
        p_61(g)=0.0
        p_70(g)=0.0
        p_53(g)=0.0
  do i=0,g
    if (i==0) then

n_out(i)=(1.0/(4.0*pi))*(2.0*(2.0*(p_10(1)/k(1))-(p_10(2)/k(2)))**2.0+2.0*(2.0*(p_20(1)/k(
1))-(p_20(2)/k(2)))**2.0+6.0*(2.0*(p_21(1)/k(1))-(p_21(2)/k(2)))**2.0&
```

```
&+2.0*(2.0*(p_30(1)/k(1))-(p_30(2)/k(2)))**2.0+6.0*(2.0*(p_31(1)/k(1))-(p_31(2)/k(2)))**2
.0+2.0*(2.0*(p_40(1)/k(1))-(p_40(2)/k(2)))**2.0&
&+10.0*(2.0*(p_32(1)/k(1))-(p_32(2)/k(2)))**2.0+6.0*(2.0*(p_41(1)/k(1))-(p_41(2)/k(2)))**
2.0+2.0*(2.0*(p_50(1)/k(1))-(p_50(2)/k(2)))**2.0&
&+10.0*(2.0*(p_42(1)/k(1))-(p_42(2)/k(2)))**2.0+6.0*(2.0*(p_51(1)/k(1))-(p_51(2)/k(2)))**
2.0+2.0*(2.0*(p_60(1)/k(1))-(p_60(2)/k(2)))**2.0&
&+10.0*(2.0*(p_52(1)/k(1))-(p_52(2)/k(2)))**2.0+14.0*(2.0*(p_43(1)/k(1))-(p_43(2)/k(2)))*
*2.0+6.0*(2.0*(p_61(1)/k(1))-(p_61(2)/k(2)))**2.0&
&+2.0*(2.0*(p_70(1)/k(1))-(p_70(2)/k(2)))**2.0+0.0*(2.0*(p_62(1)/k(1))-(p_62(2)/k(2)))**2
.0+14.0*(2.0*(p_53(1)/k(1))-(p_53(2)/k(2)))**2.0)

     else

n_out(i)=(1.0/(4.0*pi))*(2.0*(p_10(i)/k(i))**2.0+2.0*(p_20(i)/k(i))**2.0+6.0*(p_21(i)/k(i))*
*2.0+2.0*(p_30(i)/k(i))**2.0+6.0*(p_31(i)/k(i))**2.0&
&+2.0*(p_40(i)/k(i))**2.0+10.0*(p_32(i)/k(i))**2.0+6.0*(p_41(i)/k(i))**2.0+2.0*(p_50(i)/k(
i))**2.0+10.0*(p_42(i)/k(i))**2.0&
&+6.0*(p_51(i)/k(i))**2.0+2.0*(p_60(i)/k(i))**2.0+10.0*(p_52(i)/k(i))**2.0+14.0*(p_43(i)/
k(i))**2.0+6.0*(p_61(i)/k(i))**2.0&
&+2.0*(p_70(i)/k(i))**2.0+0.0*(p_62(i)/k(i))**2.0+14.0*(p_53(i)/k(i))**2.0 )
   end if
 end do
end

subroutine computing_ev_evt_ps()
use toole                        !計算 eigenvalue and eigenvector of ps.
implicit none
integer:: i,j
do j=1,g-1      !離散化之後產生的矩陣
 do i=1,g-1
   if (i==j) then
      a(i,j)=(1.0/(h**2.0)+v11_ps(i))
    else if (abs(i-j)==1) then
      a(i,j)=(-1.0/(2.0*h**2.0))
    else
      a(i,j)=0.0
   end if
  end do
end do
```

```
  do i=1,g-1
       w(i)=a(i,i)
  end do

  do i=2,g-1
       e(i)=(-1.0/(2.0*h**2.0))
  end do
call rst(g-1,g-1,w,e,1,z,ierr)

end subroutine



real*8 function adjust_c_l(x,p_nl,en)
use toole
implicit none
real(kind=8) x,en
real(kind=8) p_nl(0:g)
c_l=x
call sub15()                    !computing v_l
call cutoff_radius(p_nl,r_c) !computing r_c
call sub16()                    !v1_ps
call computing_ev_evt_ps()
adjust_c_l=w(1)-en
return
end



subroutine find_c_l01() !find c_l
use toole
implicit none
real*8,external::adjust_c_l

real(kind=8)::a1,b1,c1
real(kind=8)::fa,fb,fc
a1=-2.0
b1=0.0
fa=adjust_c_l(a1,pnl,e_nl)
```

```
fb=adjust_c_l(b1,pnl,e_nl)
c1=a1-fa*(b1-a1)/(fb-fa)
fc=adjust_c_l(c1,pnl,e_nl)
do while(abs(fc)>1e-4)
a1=b1
b1=c1
fa=adjust_c_l(a1,pnl,e_nl)
fb=adjust_c_l(b1,pnl,e_nl)
c1=a1-fa*(b1-a1)/(fb-fa)
fc=adjust_c_l(c1,pnl,e_nl)
end do
c_l=c1
end



subroutine sub15()
use toole
implicit none
integer i
  do i=1,g-1
    v_l(i)=(l*(l+1.0)/(2.0*k(i)**2.0))+v_eff(i)
  end do
end

subroutine sub16()
use toole
implicit none
integer i
real*8,external::func
    do i=1,g-1

            v1l_ps(i)=(1.0-func(k(i)/r_c))*v_l(i)+c_l*func(k(i)/r_c)

  end do

end

real*8 function    func(x)
```

```
use toole
implicit none
real(kind=8) x

   func=exp(-x**4.0)

return
end

real*8 function    gl(x)
use toole
implicit none
real(kind=8) x

   gl=x**(l+1)*exp(-x**4.0)

return
end

subroutine sub_w2_0()
use toole
implicit none
integer i
real*8,external::gl
  do i=1,g-1
    w20(i)=r1*(w10(i)+delta_l*gl(k(i)/r_c))
  end do
end

subroutine find_delta_l()
use imsl
use toole
implicit none
real(kind=8) p(3),temp1(0:g),temp2(0:g),temp3(0:g)
complex(kind=8) r(2)
integer i
real,external::trape_integral4,gl
  do i=0,g
    temp1(i)=(w10(i))**2.0
```

```
   end do
p(1)=r1**2.0*trape_integral4(temp1,h,0)-1.0


   do i=0,g
      temp2(i)=w10(i)*gl(k(i)/r_c)
   end do
p(2)=2.0*r1**2.0*trape_integral4(temp2,h,0)

   do i=0,g
      temp3(i)=(gl(k(i)/r_c))**2.0
   end do
p(3)=r1**2.0*trape_integral4(temp3,h,0)

call dzplrc(2,p,r)
delta_l=r(1)
end




subroutine cutoff_radius(p_nl,rc)
use toole
implicit none
integer,external::sequential_search
real(kind=8) P_nl(0:g),rc,temp,b(0:g)
integer i,j
b=p_nl
   do j=1,g-1
      if (abs(b(j))>abs(b(j+1))) then
         temp=b(j)
         b(j)=b(j+1)
         b(j+1)=temp
      end if
   end do
r_c=0.56*k(sequential_search(p_nl,b(g)))

end
```

```
integer function sequential_search(in,value)!循序搜尋法
use toole
implicit none
real(kind=8) value
real(kind=8) in(0:g)
integer i

do i=0,g
        if (value==in(i)) then
              sequential_search=i
        return
        end if
end do
sequential_search=0
return

end function




subroutine pseudo()
use toole
implicit none
integer i
do i=1,g-1
v2l_ps(i)=(e_nl*w20(i)+0.5*((w20(i+1)-2.0*w20(i)+w20(i-1))/h**2.0))*(1.0/w20(i))
end do
end

subroutine E_G() !total energy 計算
use toole
implicit none
call sub3()
call KE()     !動能 T
n=n_out
call radius()    !the radius r_s of a unit charge
```

```
call ee()           !the exchange energy e_x
call ce()           !the correlation energy    e_c
call ece()          !the exchange-correlation energy per unit density e_xc
call sub4()
call Excfunctional()    !exchange-correlation fuctional E_xc
call sub5()
call sub6()                     !計算總能中的 ∫v(r)n(r)dr
call sub7()
call sub8()
call sub9()                     !!coulomb energy (1/2)*∫∫(Vee*n)dr
total_energy=kinetic_energy+E_xc_functional+external_energy+coulomb_energy
end


!find the eigenvalues and eigenvectors (if desired)
!of a real symmetric tridiagonal matrix.
subroutine rst(nm,n,w,e,matz,z,ierr)
integer i,j,n,nm,ierr,matz
double precision w(n),e(n),z(nm,n)
!c
!c        this subroutine calls the recommended sequence of
!c        subroutines from the eigensystem subroutine package (eispack)
!c        to find the eigenvalues and eigenvectors (if desired)
!c        of a real symmetric tridiagonal matrix.
!c
!c        on input
!c
!c            nm    must be set to the row dimension of the two-dimensional
!c            array parameters as declared in the calling program
!c            dimension statement.
!c
!c            n    is the order of the matrix.
!c
!c            w    contains the diagonal elements of the real
!c            symmetric tridiagonal matrix.
!c
!c            e    contains the subdiagonal elements of the matrix in
!c            its last n-1 positions.    e(1) is arbitrary.
!c
```

```
!c             matz    is an integer variable set equal to zero if
!c             only eigenvalues are desired.    otherwise it is set to
!c             any non-zero integer for both eigenvalues and eigenvectors.
!c
!c        on output
!c
!c
!c             z    contains the eigenvectors if matz is not zero.
!c
!c             ierr    is an integer output variable set equal to an error
!c                  completion code described in the documentation for imtql1
!c                  and imtql2.    the normal completion code is zero.
!c
!c        questions and comments should be directed to burton s. garbow,
!c        mathematics and computer science div, argonne national laboratory
!c
!c        this version dated august 1983.
!c
!c        ------------------------------------------------------------------
          if (n .le. nm) go to 10
          ierr = 10 * n
          go to 50
!c
     10 if (matz .ne. 0) go to 20
!c        .......... find eigenvalues only ..........
          call    imtql1(n,w,e,ierr)
          go to 50
!c        .......... find both eigenvalues and eigenvectors ..........
     20 do 40 i = 1, n
!c
             do 30 j = 1, n
                z(j,i) = 0.0d0
     30      continue
!c
             z(i,i) = 1.0d0
     40 continue
!c
          call    imtql2(nm,n,w,e,z,ierr)
     50 return
```

```
end



subroutine imtql1(n,d,e,ierr)
integer i,j,l,m,n,ii,mml,ierr
double precision d(n),e(n)
double precision b,c,f,g,p,r,s,tst1,tst2,pythag
!c
!c      this subroutine is a translation of the algol procedure imtql1,
!c      num. math. 12, 377-383(1968) by martin and wilkinson,
!c      as modified in num. math. 15, 450(1970) by dubrulle.
!c      handbook for auto. comp., vol.ii-linear algebra, 241-248(1971).
!c
!c      this subroutine finds the eigenvalues of a symmetric
!c      tridiagonal matrix by the implicit ql method.
!c
!c      on input
!c
!c         n is the order of the matrix.
!c
!c         d contains the diagonal elements of the input matrix.
!c
!c         e contains the subdiagonal elements of the input matrix
!c           in its last n-1 positions.    e(1) is arbitrary.
!c
!c       on output
!c
!c         d contains the eigenvalues in ascending order.    if an
!c           error exit is made, the eigenvalues are correct and
!c           ordered for indices 1,2,...ierr-1, but may not be
!c           the smallest eigenvalues.
!c
!c         e has been destroyed.
!c
!c         ierr is set to
!c           zero         for normal return,
!c           j            if the j-th eigenvalue has not been
!c                         determined after 30 iterations.
```

```
!c
!1c      calls pythag for    dsqrt(a*a + b*b) .
!c
!c       questions and comments should be directed to burton s. garbow,
!c       mathematics and computer science div, argonne national laboratory
!c
!c       this version dated august 1983.
!c
!c       ------------------------------------------------------------------
!c
         ierr = 0
         if (n .eq. 1) go to 1001
!c
         do 100 i = 2, n
  100 e(i-1) = e(i)
!c
         e(n) = 0.0d0
!c
         do 290 l = 1, n
            j = 0
!c       .......... look for small sub-diagonal element ..........
  105       do 110 m = l, n
               if (m .eq. n) go to 120
               tst1 = dabs(d(m)) + dabs(d(m+1))
               tst2 = tst1 + dabs(e(m))
               if (tst2 .eq. tst1) go to 120
  110       continue
!c
  120       p = d(l)
            if (m .eq. l) go to 215
            if (j .eq. 30) go to 1000
            j = j + 1
!c       .......... form shift ..........
            g = (d(l+1) - p) / (2.0d0 * e(l))
            r = pythag(g,1.0d0)
            g = d(m) - p + e(l) / (g + dsign(r,g))
            s = 1.0d0
            c = 1.0d0
            p = 0.0d0
```

```
        mml = m - 1
!c      .......... for i=m-1 step -1 until l do -- ..........
        do 200 ii = 1, mml
           i = m - ii
           f = s * e(i)
           b = c * e(i)
           r = pythag(f,g)
           e(i+1) = r
           if (r .eq. 0.0d0) go to 210
           s = f / r
           c = g / r
           g = d(i+1) - p
           r = (d(i) - g) * s + 2.0d0 * c * b
           p = s * r
           d(i+1) = g + p
           g = c * r - b
  200   continue
!c
        d(l) = d(l) - p
        e(l) = g
        e(m) = 0.0d0
        go to 105
!c      .......... recover from underflow ..........
  210   d(i+1) = d(i+1) - p
        e(m) = 0.0d0
        go to 105
!c      .......... order eigenvalues ..........
  215   if (l .eq. 1) go to 250
!c      .......... for i=l step -1 until 2 do -- ..........
        do 230 ii = 2, l
           i = l + 2 - ii
           if (p .ge. d(i-1)) go to 270
           d(i) = d(i-1)
  230   continue
!c
  250   i = 1
  270   d(i) = p
  290 continue
!c
```

```
      go to 1001
!c      .......... set error -- no convergence to an
!c                 eigenvalue after 30 iterations ..........
 1000 ierr = l
 1001 return
end


subroutine imtql2(nm,n,d,e,z,ierr)
integer i,j,k,l,m,n,ii,nm,mml,ierr
double precision d(n),e(n),z(nm,n)
double precision b,c,f,g,p,r,s,tst1,tst2,pythag
!c
!c      this subroutine is a translation of the algol procedure imtql2,
!c      num. math. 12, 377-383(1968) by martin and wilkinson,
!c      as modified in num. math. 15, 450(1970) by dubrulle.
!c      handbook for auto. comp., vol.ii-linear algebra, 241-248(1971).
!c
!c      this subroutine finds the eigenvalues and eigenvectors
!c      of a symmetric tridiagonal matrix by the implicit ql method.
!c      the eigenvectors of a full symmetric matrix can also
!c      be found if  tred2  has been used to reduce this
!c      full matrix to tridiagonal form.
!c
!c      on input
!c
!c         nm must be set to the row dimension of two-dimensional
!c           array parameters as declared in the calling program
!c           dimension statement.
!c
!c         n is the order of the matrix.
!c
!c         d contains the diagonal elements of the input matrix.
!c
!c         e contains the subdiagonal elements of the input matrix
!c           in its last n-1 positions.  e(1) is arbitrary.
!c
!c         z contains the transformation matrix produced in the
!c           reduction by  tred2, if performed.  if the eigenvectors
```

```
!c              of the tridiagonal matrix are desired, z must contain
!c              the identity matrix.
!c
!c        on output
!c
!c           d contains the eigenvalues in ascending order.   if an
!c              error exit is made, the eigenvalues are correct but
!c              unordered for indices 1,2,...,ierr-1.
!c
!c           e has been destroyed.
!c
!c           z contains orthonormal eigenvectors of the symmetric
!c              tridiagonal (or full) matrix.   if an error exit is made,
!c              z contains the eigenvectors associated with the stored
!c              eigenvalues.
!c
!c           ierr is set to
!c              zero        for normal return,
!c              j           if the j-th eigenvalue has not been
!c                          determined after 30 iterations.
!c
!c       calls pythag for   dsqrt(a*a + b*b) .
!c
!c       questions and comments should be directed to burton s. garbow,
!c       mathematics and computer science div, argonne national laboratory
!c
!c       this version dated august 1983.
!c
!c       ------------------------------------------------------------------
!c
      ierr = 0
      if (n .eq. 1) go to 1001
!c
      do 100 i = 2, n
  100 e(i-1) = e(i)
!c
      e(n) = 0.0d0
!c
      do 240 l = 1, n
```

```
          j = 0
!c     .......... look for small sub-diagonal element ..........
  105     do 110 m = l, n
             if (m .eq. n) go to 120
             tst1 = dabs(d(m)) + dabs(d(m+1))
             tst2 = tst1 + dabs(e(m))
             if (tst2 .eq. tst1) go to 120
  110     continue
!c
  120     p = d(l)
          if (m .eq. l) go to 240
          if (j .eq. 30) go to 1000
          j = j + 1
!c     .......... form shift ..........
          g = (d(l+1) - p) / (2.0d0 * e(l))
          r = pythag(g,1.0d0)
          g = d(m) - p + e(l) / (g + dsign(r,g))
          s = 1.0d0
          c = 1.0d0
          p = 0.0d0
          mml = m - l
!c     .......... for i=m-1 step -1 until l do -- ..........
          do 200 ii = 1, mml
             i = m - ii
             f = s * e(i)
             b = c * e(i)
             r = pythag(f,g)
             e(i+1) = r
             if (r .eq. 0.0d0) go to 210
             s = f / r
             c = g / r
             g = d(i+1) - p
             r = (d(i) - g) * s + 2.0d0 * c * b
             p = s * r
             d(i+1) = g + p
             g = c * r - b
!c     .......... form vector ..........
             do 180 k = 1, n
                f = z(k,i+1)
```

```
                z(k,i+1) = s * z(k,i) + c * f
                z(k,i) = c * z(k,i) - s * f
  180           continue
!c
  200       continue
!c
            d(l) = d(l) - p
            e(l) = g
            e(m) = 0.0d0
            go to 105
!c        .......... recover from underflow ..........
  210       d(i+1) = d(i+1) - p
            e(m) = 0.0d0
            go to 105
  240 continue
!c        .......... order eigenvalues and eigenvectors ..........
        do 300 ii = 2, n
            i = ii - 1
            k = i
            p = d(i)
!c
            do 260 j = ii, n
                if (d(j) .ge. p) go to 260
                k = j
                p = d(j)
  260       continue
!c
            if (k .eq. i) go to 300
            d(k) = d(i)
            d(i) = p
!c
            do 280 j = 1, n
                p = z(j,i)
                z(j,i) = z(j,k)
                z(j,k) = p
  280       continue
!!c
  300 continue
!c
```

```
          go to 1001
!c          .......... set error -- no convergence to an
!c                          eigenvalue after 30 iterations ..........
 1000 ierr = l
 1001 return
end


double precision function pythag(a,b)
double precision a,b
!c
!c          finds dsqrt(a**2+b**2) without overflow or destructive underflow
!c
          double precision p,r,s,t,u
          p = dmax1(dabs(a),dabs(b))
          if (p .eq. 0.0d0) go to 20
          r = (dmin1(dabs(a),dabs(b))/p)**2
     10 continue
               t = 4.0d0 + r
               if (t .eq. 4.0d0) go to 20
               s = r/t
               u = 1.0d0 + 2.0d0*s
               p = u*p
               r = (s/u)**2 * r
          go to 10
     20 pythag = p
          return
end
```

A.2 The initial electron density of No atom(1*401)

107084.007828308	34372.7202003888	5300.30345368854
2745.91161917885	1624.66343877385	892.197653574079
517.051763519848	354.800421382601	280.003233664372
231.320654298285	188.341687507305	148.569806588318
113.740401626887	85.3953312399476	63.6774371773841
47.8322918811604	36.6624632645909	28.9606046635084
23.6807816232239	20.0194356502229	17.3967305601694
15.4231953497467	13.8477810162582	12.5191465439582
11.3492259742482	10.2911567867160	9.32069403295397
8.42701541242964	7.60486812302368	6.85187952990541
6.16574673671114	5.54406555171051	4.98350001575553
4.48028895432457	4.03005286824163	3.62830563879969
3.27040769438993	2.95191799334773	2.66854854916613
2.41636727839534	2.19173325135725	1.99139802790508
1.81243131970336	1.65226375522610	1.50861503142306
1.37950656016855	1.26319882912490	1.15819069653529
1.06316840210656	0.976999821221959	0.898695023259516
0.827399583719429	0.762364868721077	0.702942201908803
0.648560898054416	0.598723661220932	0.552990586063213
0.510975795157975	0.472335824607880	0.436767214016821
0.403998090272881	0.373786537892698	0.345914497227430
0.320186606536367	0.296425802473405	0.274472443596859
0.254181144073415	0.235420159665273	0.218068965152489
0.202017856902356	0.187166163485957	0.173421923097319
0.160700544202954	0.148924491324130	0.138022306315634
0.127928344025784	0.118582005664314	0.109927520435997
0.101913332498783	9.449192099516436E-002	8.761932895483111E-002
8.125498364289073E-002	7.536132896241667E-002	6.990367048803577E-002
6.484987503584268E-002	6.017024046721889E-002	5.583724402556516E-002
5.182542753444918E-002	4.811119642185761E-002	4.467271542712876E-002
4.148974157580070E-002	3.854353246913057E-002	3.581671468731099E-002
3.329319502591192E-002	3.095805076599340E-002	2.879745787173793E-002
2.679859747752530E-002	2.494959032376140E-002	2.323941942108340E-002
2.165787433951473E-002	2.019548439481859E-002	1.884347232932811E-002
1.759369667040717E-002	1.643860878903685E-002	1.537120840670903E-002
1.438500497240101E-002	1.347397809970459E-002	1.263254693531655E-002

1.185553407455260E-002	1.113813960804344E-002	1.047591147786275E-002
9.864722160225423E-003	9.300743863008295E-003	8.780427516106309E-003
8.300481899761514E-003	7.857856086626652E-003	7.449721203108031E-003
7.073454903081042E-003	6.726625618606037E-003	6.406980137083121E-003
6.112429498216537E-003	5.841038107483337E-003	5.591011940225279E-003
5.360689047079300E-003	5.148529188640743E-003	4.953105720501421E-003
4.773097140783451E-003	4.607279767040080E-003	4.454520103278698E-003
4.313768928726015E-003	4.184054900614255E-003	4.064479111616121E-003
3.954209759697231E-003	3.852477476473913E-003	3.758570222046024E-003
3.671829862411742E-003	3.591648038451772E-003	3.517462724052065E-003
3.448754308609661E-003	3.385043110725487E-003	3.325886409134098E-003
3.270875381509599E-003	3.219633054924623E-003	3.171811762731331E-003
3.127091196957642E-003	3.085176423897589E-003	3.045795870980201E-003
3.008699695388421E-003	2.973658235972852E-003	2.940460565578269E-003
2.908913158824198E-003	2.878838408990432E-003	2.850073623823818E-003
2.822470043278500E-003	2.795891543766766E-003	2.770213858375975E-003
2.745323703078859E-003	2.721117957650124E-003	2.697502897497303E-003
2.674393472946103E-003	2.651712573239848E-003	2.629390629829836E-003
2.607364923534576E-003	2.585579058293296E-003	2.563982437714587E-003
2.542529953206961E-003	2.521181418041916E-003	2.499901211479095E-003
2.478658027306466E-003	2.457424453632664E-003	2.436176558278047E-003
2.414893696350247E-003	2.393558203744460E-003	2.372155294234838E-003
2.350672575447646E-003	2.329100071057797E-003	2.307429861672135E-003
2.285656019340413E-003	2.263774214852109E-003	2.241781791908709E-003
2.219677495588686E-003	2.197461331112933E-003	2.175134536064552E-003
2.152699472602142E-003	2.130159293665480E-003	2.107518102411886E-003
2.084780655473636E-003	2.061952400499435E-003	2.039039288663764E-003
2.016047870133048E-003	1.992985135609288E-003	1.969858403951470E-003
1.946675281901410E-003	1.923443632740960E-003	1.900171567406406E-003
1.876867266208779E-003	1.853539117905898E-003	1.830195594001334E-003
1.806845158029764E-003	1.783496274244414E-003	1.760157467748888E-003
1.736837183780744E-003	1.713543843074242E-003	1.690285828883776E-003
1.667071255161842E-003	1.643908202517696E-003	1.620804612982832E-003
1.597768324217861E-003	1.574806982149893E-003	1.551928041618833E-003
1.529138801069894E-003	1.506446337823151E-003	1.483857526295779E-003
1.461379109343330E-003	1.439017530768938E-003	1.416779063764879E-003
1.394669765906052E-003	1.372695530528835E-003	1.350861920173128E-003
1.329174369132426E-003	1.307638102807593E-003	1.286258039648673E-003

1.265038930184872E-003	1.243985324631714E-003	1.223101534021123E-003
1.202391640081171E-003	1.181859535426908E-003	1.161508907636419E-003
1.141343220490764E-003	1.121365722369299E-003	1.101579499595279E-003
1.081987414842048E-003	1.062592179433795E-003	1.043396265656977E-003
1.024401978048282E-003	1.005611430924838E-003	9.870265691125674E-004
9.686491649160720E-004	9.504808334294193E-004	9.325230324215833E-004
9.147770038020224E-004	8.972439105427312E-004	8.799247079612382E-004
8.628202102971768E-004	8.459311125501529E-004	8.292579726991195E-004
8.128011744773198E-004	7.965610474378127E-004	7.805377097853968E-004
7.647312076441417E-004	7.491414647760732E-004	7.337682930893217E-004
7.186113914133056E-004	7.036703658639563E-004	6.889447092259839E-004
6.744338387625802E-004	6.601370304849589E-004	6.460535266826204E-004
6.321824790771983E-004	6.185229287273050E-004	6.050738387382229E-004
5.918341191080150E-004	5.788026139460014E-004	5.659780795942923E-004
5.533592114626592E-004	5.409446700569664E-004	5.287330122818653E-004
5.167227890248111E-004	5.049124808675147E-004	4.933004954917013E-004
4.818852130320150E-004	4.706649924409126E-004	4.596381181057025E-004
4.488028633045277E-004	4.381574369243010E-004	4.277000418730818E-004
4.174288444855190E-004	4.073419754864941E-004	3.974375490586837E-004
3.877136591572628E-004	3.781683578037545E-004	3.687996739624399E-004
3.596056479205088E-004	3.505842764935176E-004	3.417335532411549E-004
3.330514570944315E-004	3.245359523655750E-004	3.161850116181944E-004
3.079965831750256E-004	2.999686163511237E-004	2.920990624907081E-004
2.843858490141192E-004	2.768269115816597E-004	2.694201843353941E-004
2.621636143384974E-004	2.550551277448907E-004	2.480926721292887E-004
2.412741936935813E-004	2.345976469758367E-004	2.280609911969619E-004
2.216621930621292E-004	2.153992161898536E-004	2.092700470485398E-004
2.032726756341046E-004	1.974051051394466E-004	1.916653515765149E-004
1.860514357313390E-004	1.805613948654891E-004	1.751932807943345E-004
1.699451582707738E-004	1.648151092777054E-004	1.598012239753983E-004
1.549016118921286E-004	1.501144006286792E-004	1.454377292444565E-004
1.408697603609828E-004	1.364086660527135E-004	1.320526440699947E-004
1.277999033592902E-004	1.236486735183032E-004	1.195972041386971E-004
1.156437607345007E-004	1.117866294336247E-004	1.080241178349560E-004
1.043545499571042E-004	1.007762716294620E-004	9.728764813793271E-005
9.388706333789075E-005	9.057292307064520E-005	8.734365366187201E-005
8.419769986704480E-005	8.113352736170394E-005	7.814962404826182E-005
7.524449709139090E-005	7.241667489680423E-005	6.966470673017529E-005

6.698716124042257E-005	6.438262942915002E-005	6.184971996407868E-005
5.938706592125868E-005	5.699331877717903E-005	5.466715074595912E-005
5.240725605101484E-005	5.021234720585632E-005	4.808115892318528E-005
4.601244548432785E-005	4.400498164801167E-005	4.205756181876279E-005
4.016900113533645E-005	3.833813489737094E-005	3.656381758656045E-005
3.484492388128720E-005	3.318034686957430E-005	3.156900082257126E-005
3.000981798593975E-005	2.850175084636160E-005	2.704377065421627E-005
2.563486704568786E-005	2.427404903926884E-005	2.296034428220840E-005
2.169279860606880E-005	2.047047597764557E-005	1.929245954364334E-005
1.815784953029817E-005	1.706576500327527E-005	1.601534152541144E-005
1.500573313169347E-005	1.403611052929391E-005	1.310566181693788E-005
1.221359259022907E-005	1.135912510726964E-005	1.054149817778547E-005
9.759967616749666E-006	9.013805232616786E-006	8.302299182735209E-006
7.624753618472298E-006	6.980488950541492E-006	6.368841018787273E-006
5.789161395555947E-006	5.240817286523790E-006	4.723191124608669E-006
4.235680322660612E-006	3.777697330128666E-006	3.348669674857610E-006
2.948039405738857E-006	2.575263362181452E-006	2.229812750981608E-006
1.911173186825374E-006	1.618844395690042E-006	1.352340243898984E-006
1.111188596421449E-006	8.949311547228810E-007	7.031234266772041E-007
5.353346255180588E-007	3.911475602763726E-007	2.701585979875611E-007
1.719775726759456E-007	9.622777119175851E-008	4.254590363656890E-008
1.058210769637239E-008	0.000000000000000E+000	

A.3 The diagrams of output resulting

In these diagrams, pseudo wave functions and radial functions P_{nl} are described by red lines and green lines, respectively.

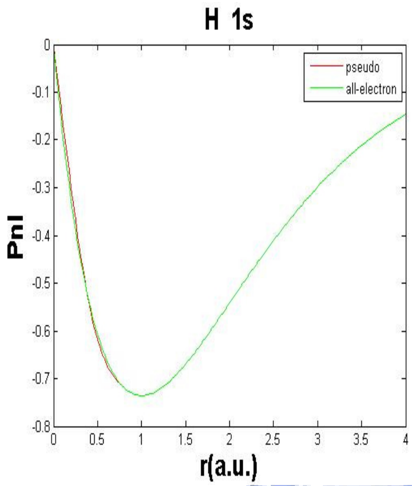


Figure 3.

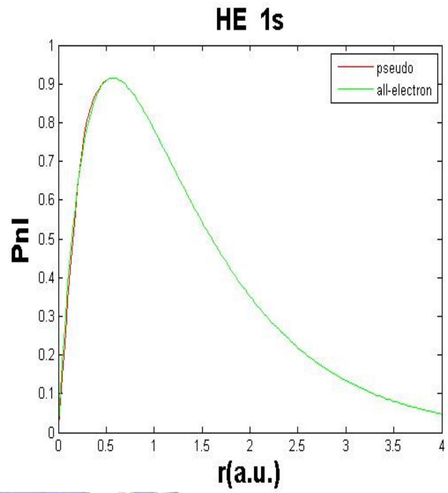


Figure 4.

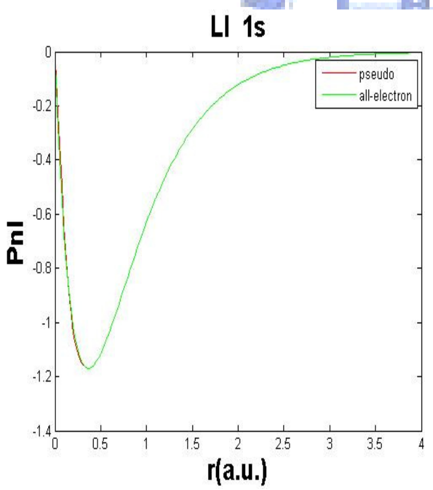


Figure 5.

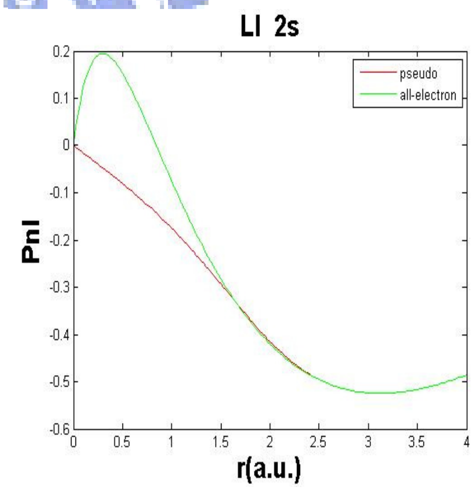


Figure 6.

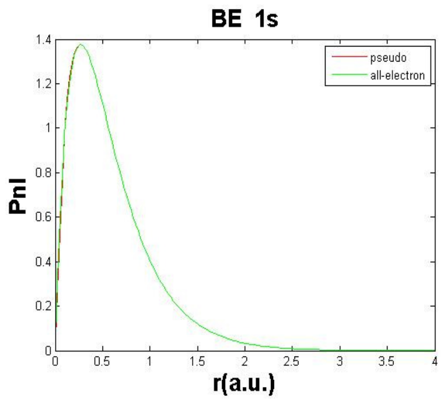


Figure 7.

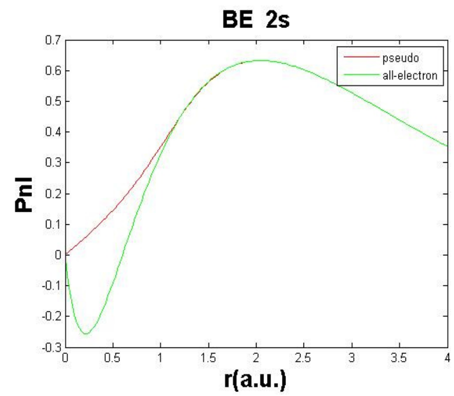


Figure 8.

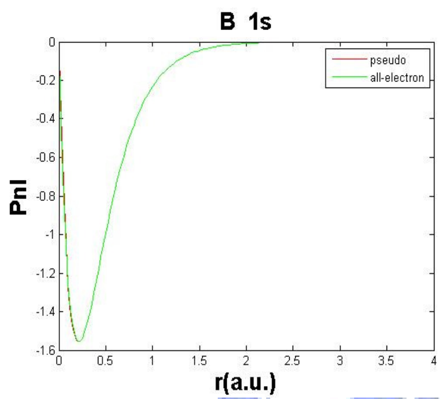


Figure 9.

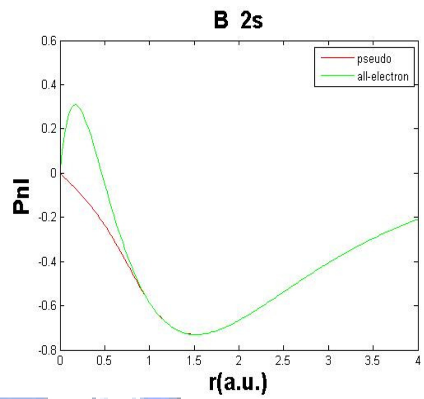


Figure 10.

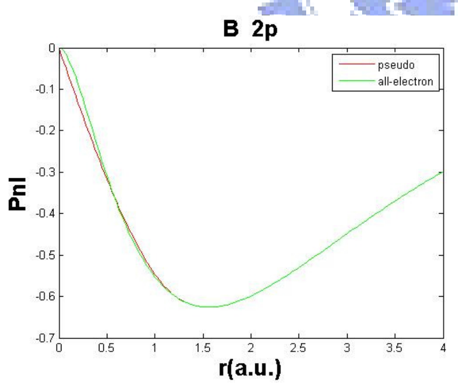


Figure 11.

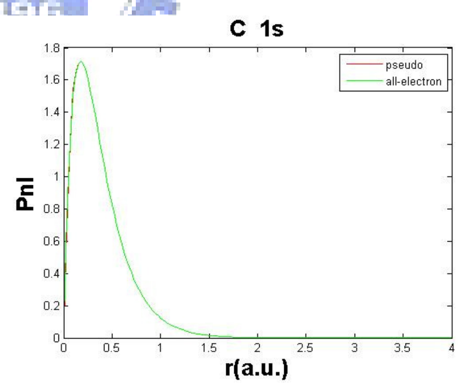


Figure 12.

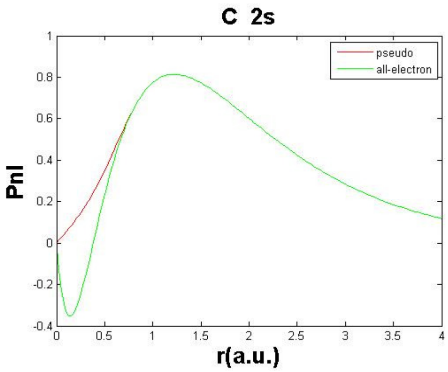


Figure 13.

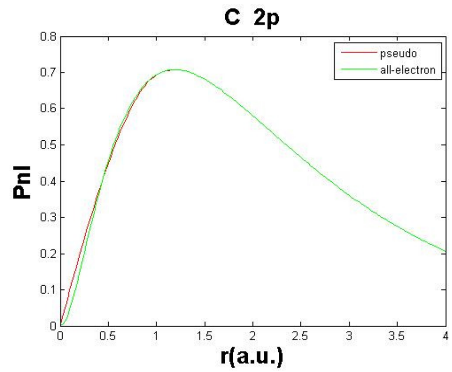


Figure 14.

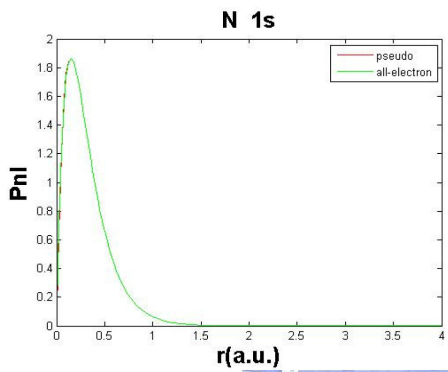


Figure 15.

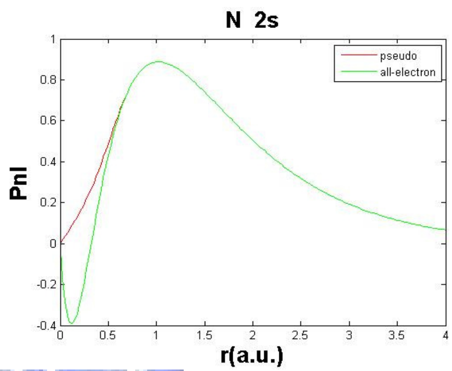


Figure 16.

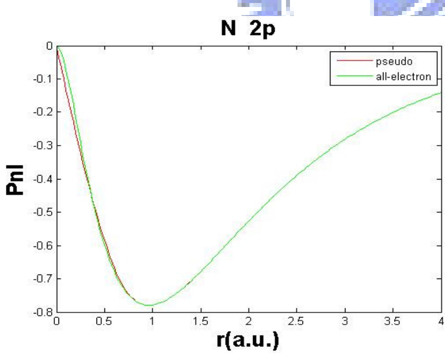


Figure 17.

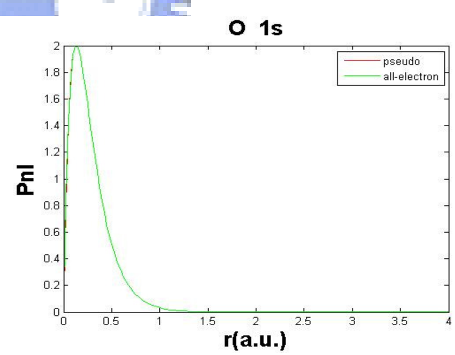


Figure 18.

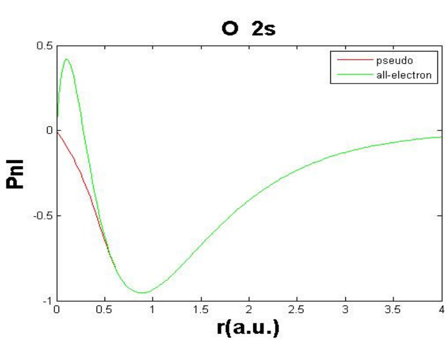


Figure 19.

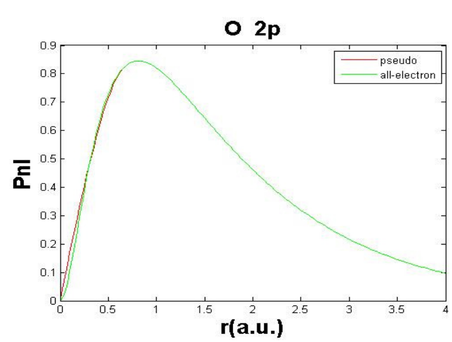


Figure 20.

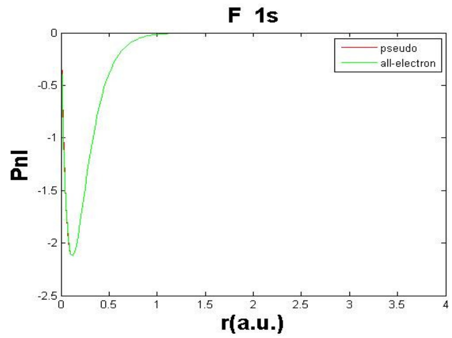


Figure 21.

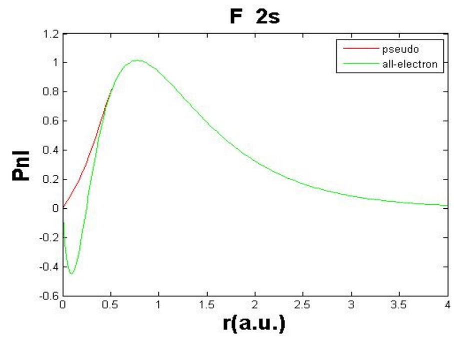


Figure 22.

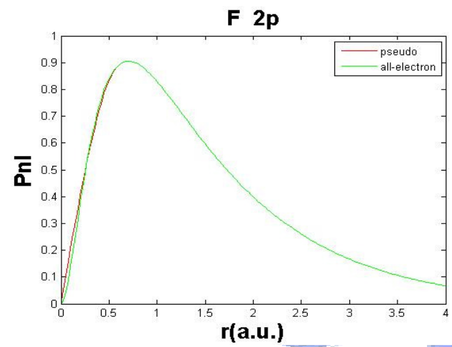


Figure 23.

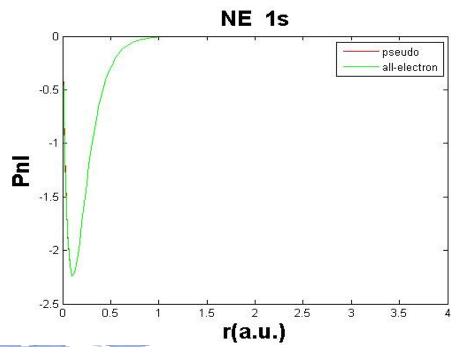


Figure 24.

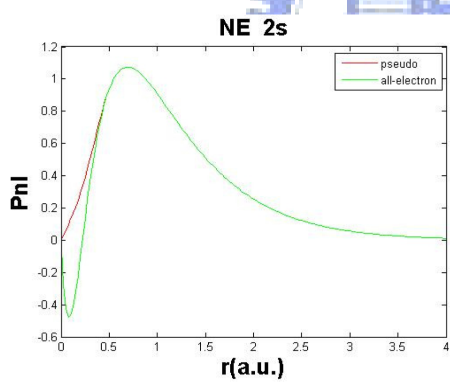


Figure 25.

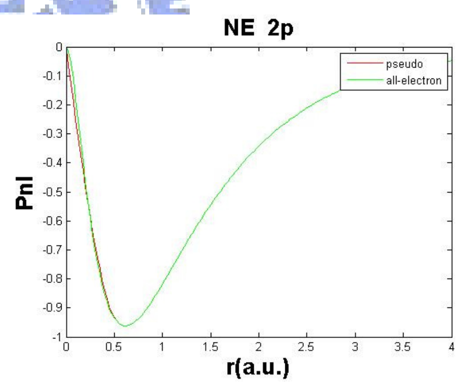


Figure 26.

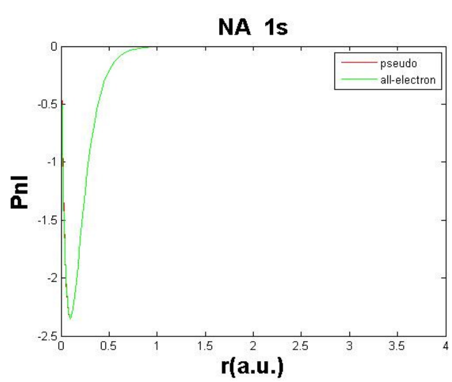


Figure 27.

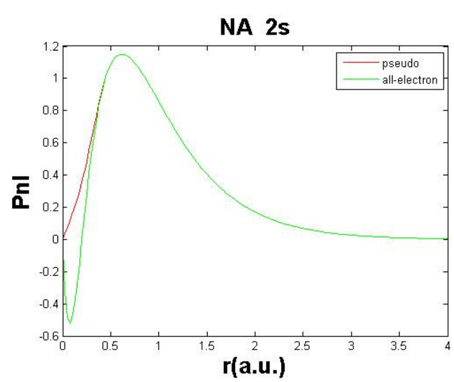


Figure 28.

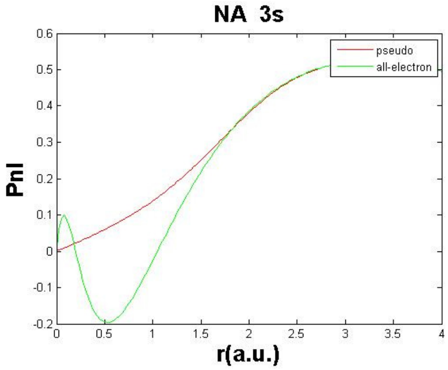


Figure 29.

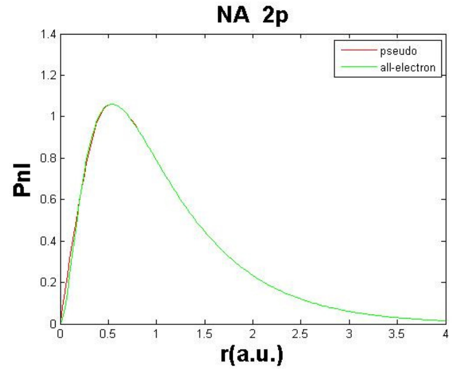


Figure 30.

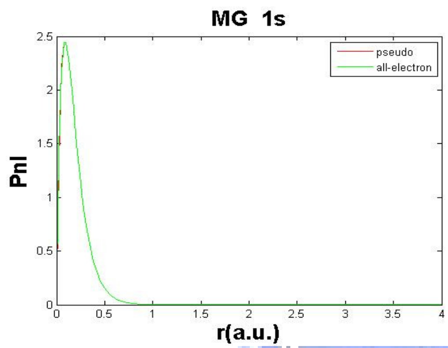


Figure 31.

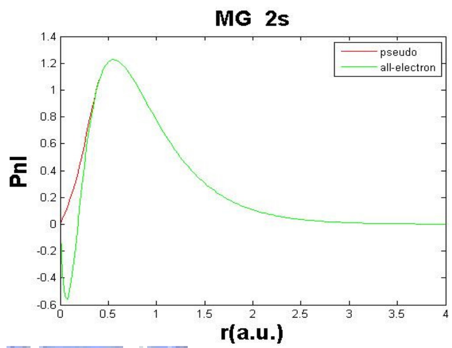


Figure 32.

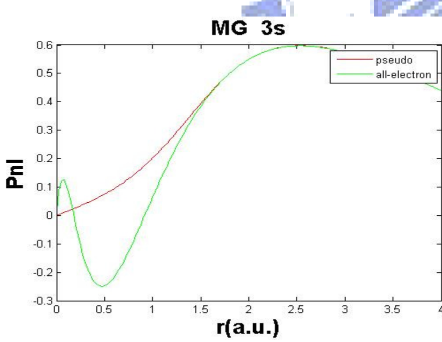


Figure 33.

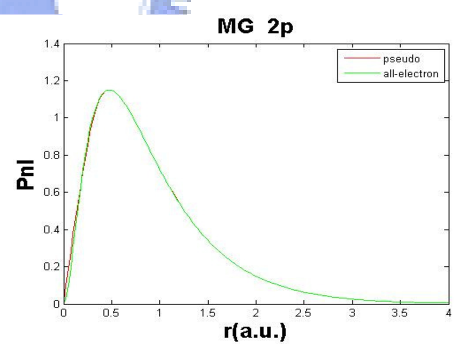


Figure 34.

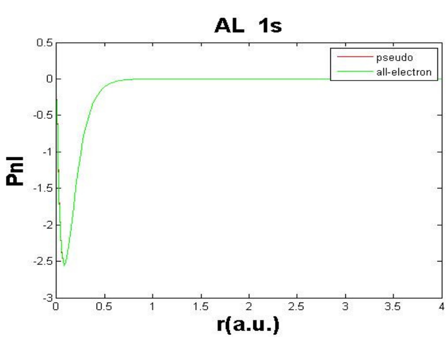


Figure 35.

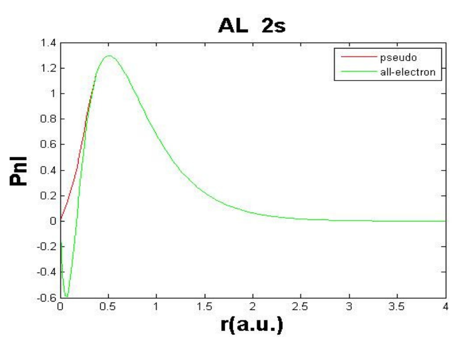


Figure 36.

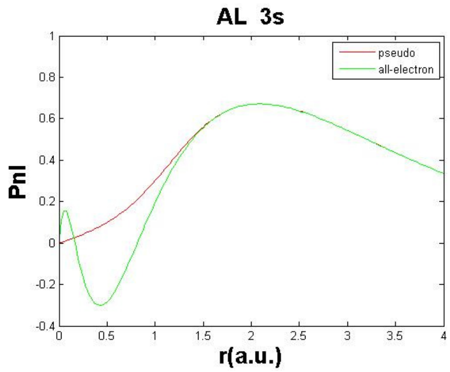


Figure 37.

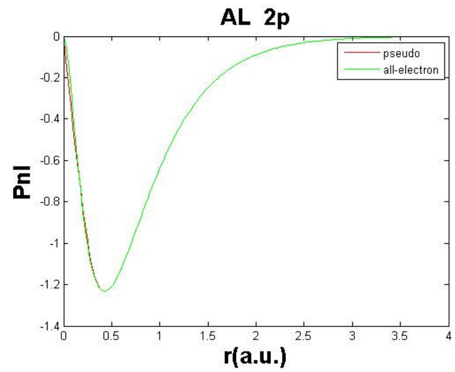


Figure 38.

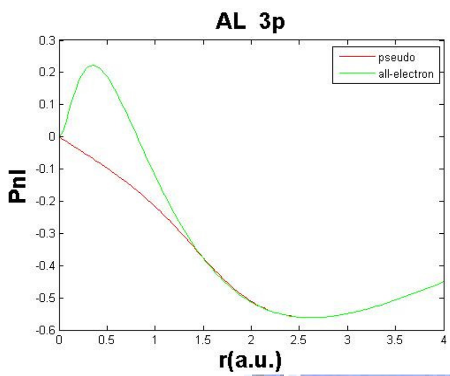


Figure 39.

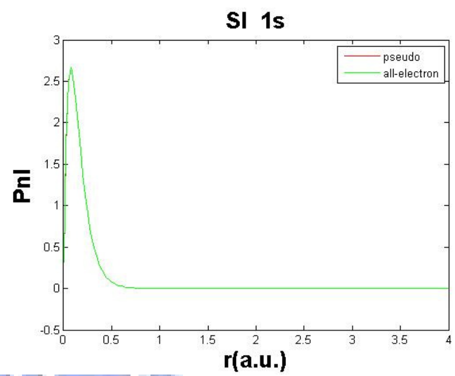


Figure 40.

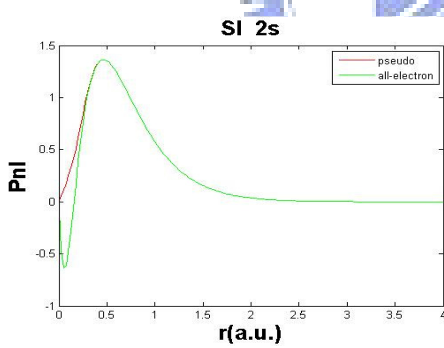


Figure 41.

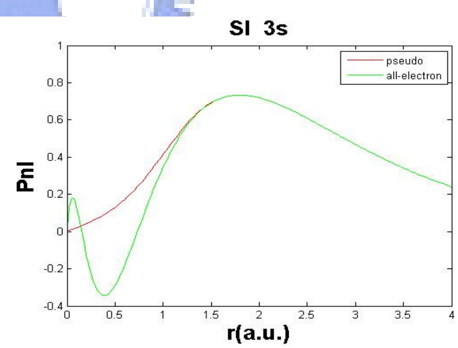


Figure 42.

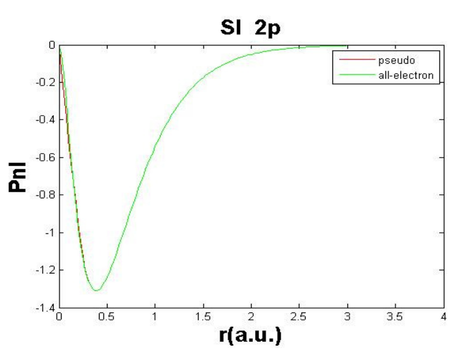


Figure 43.

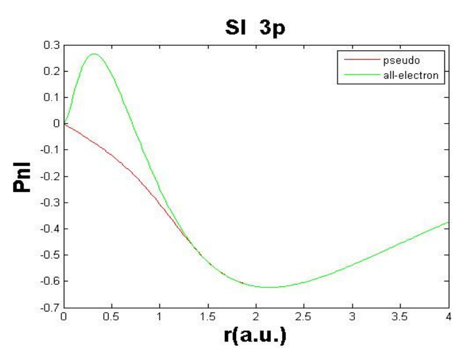


Figure 44.

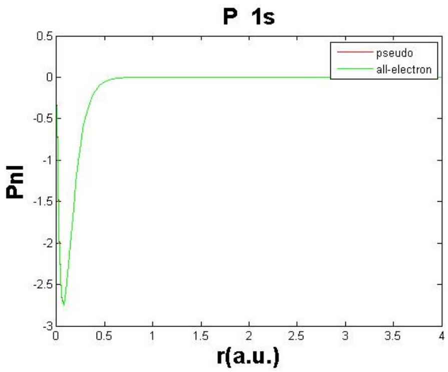


Figure 45.

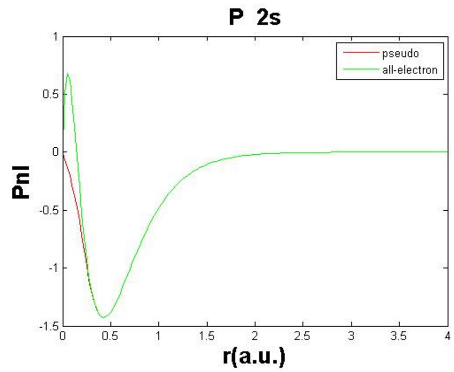


Figure 46.

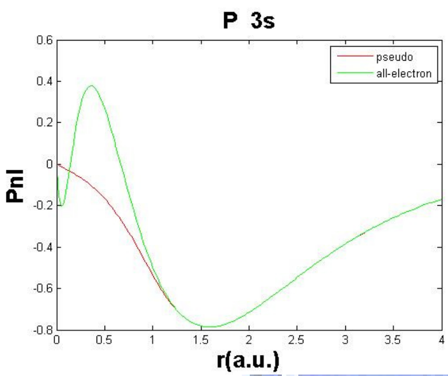


Figure 47.

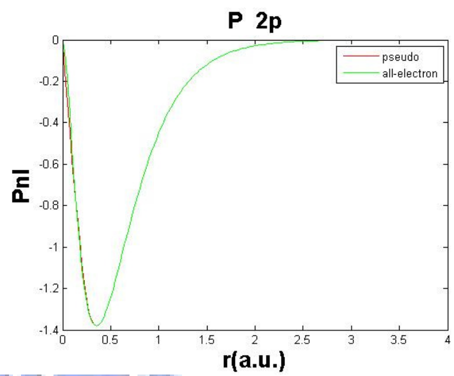


Figure 48.

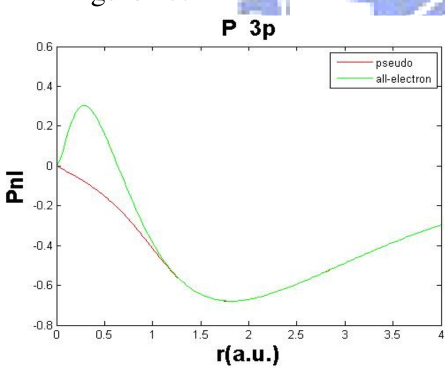


Figure 49.

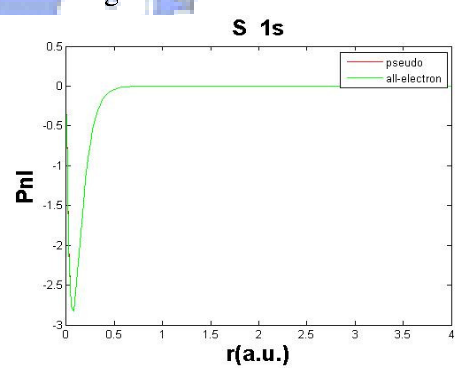


Figure 50.

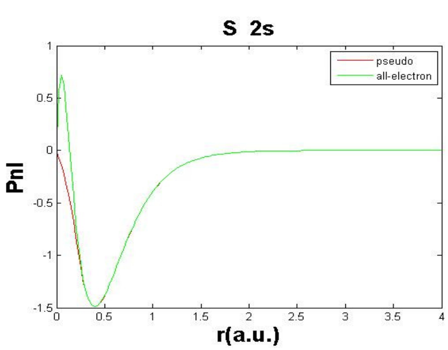


Figure 51.

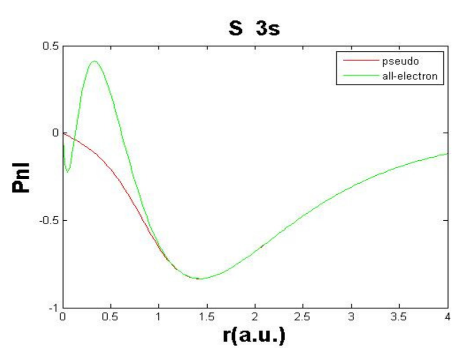


Figure 52.

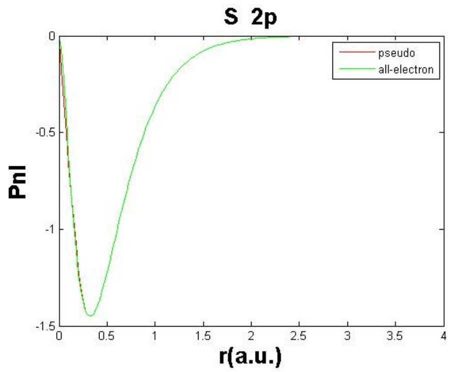


Figure 53.

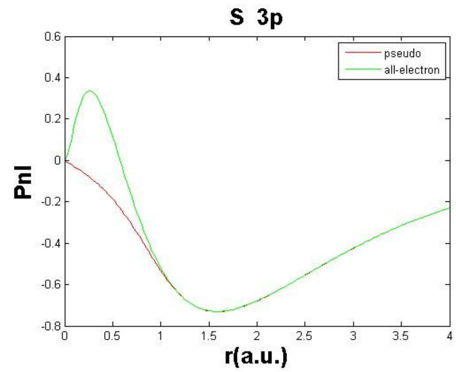


Figure 54.

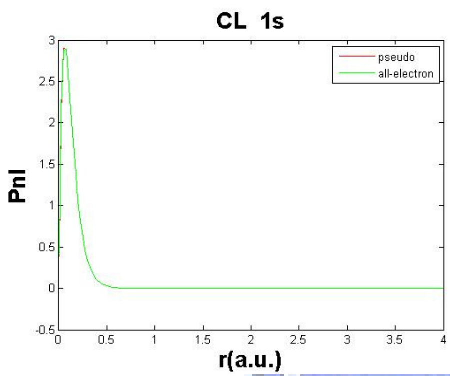


Figure 55.

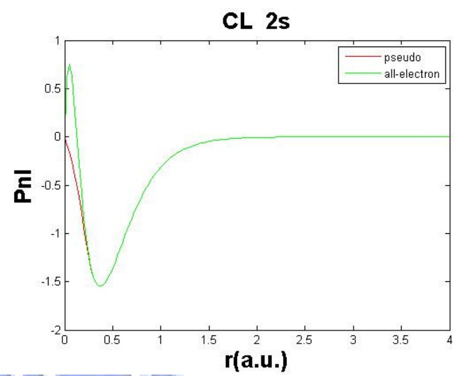


Figure 56.

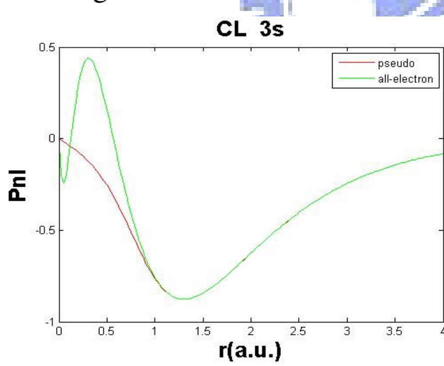


Figure 57.

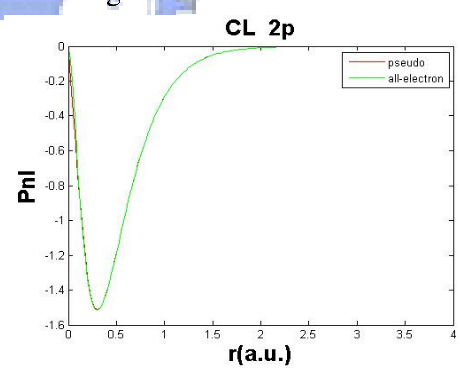


Figure 58.

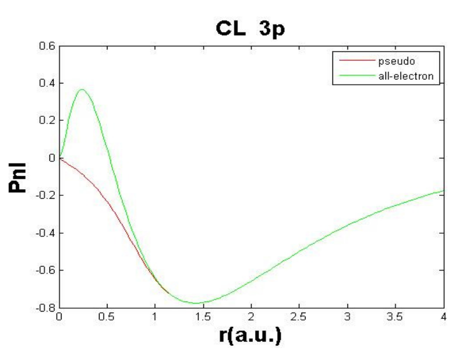


Figure 59.

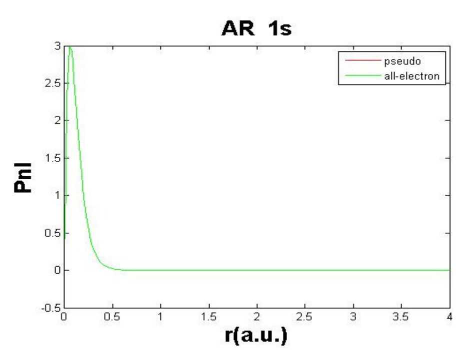


Figure 60.

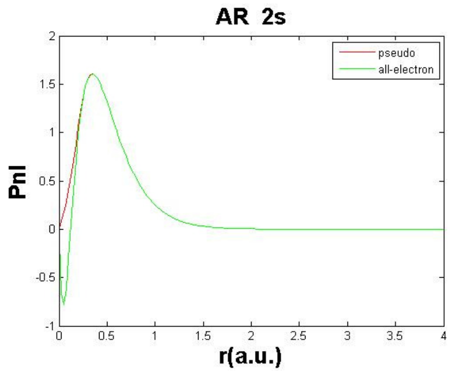


Figure 61.

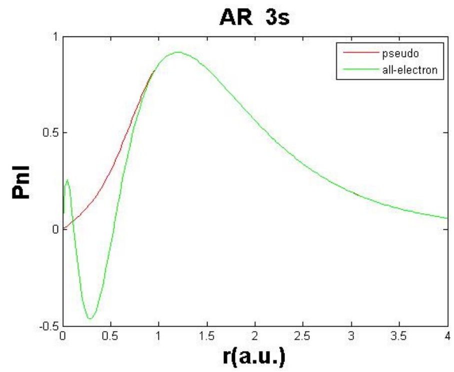


Figure 62.

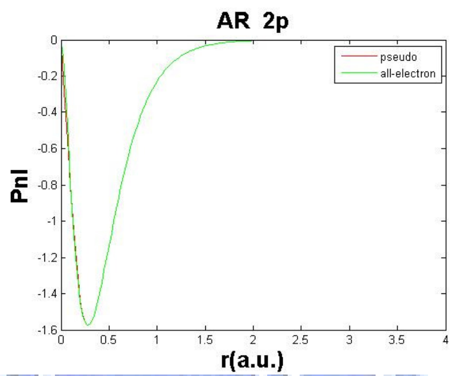


Figure 63.

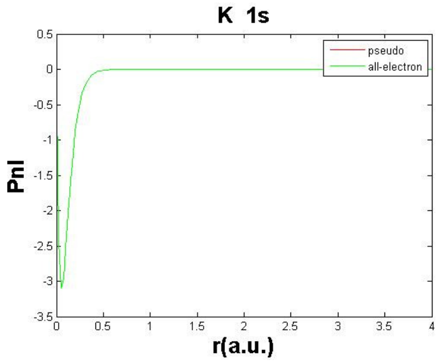


Figure 64.

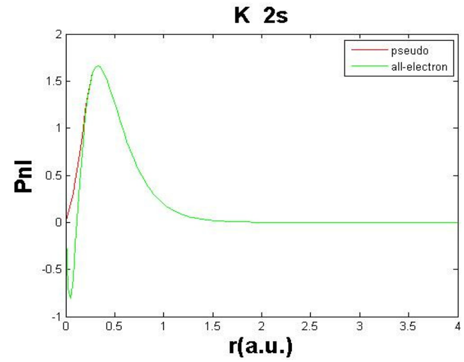


Figure 65.

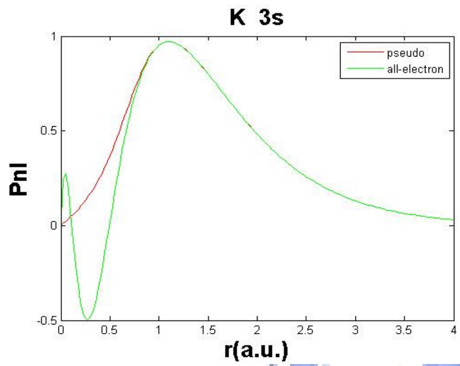


Figure 66.

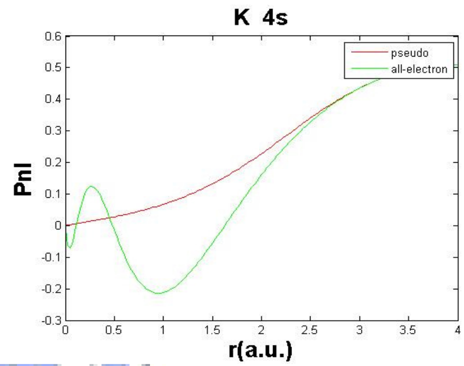


Figure 67.

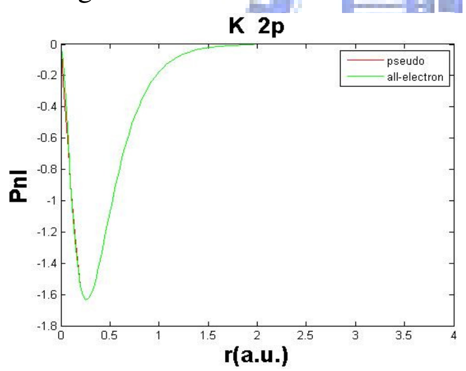


Figure 68.

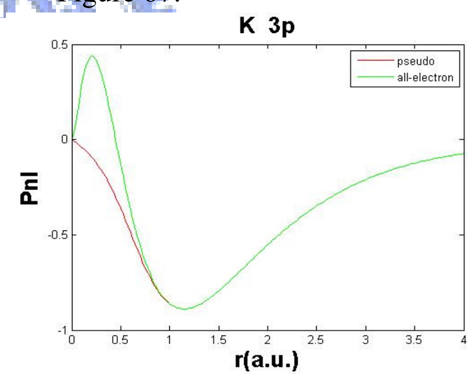


Figure 69.

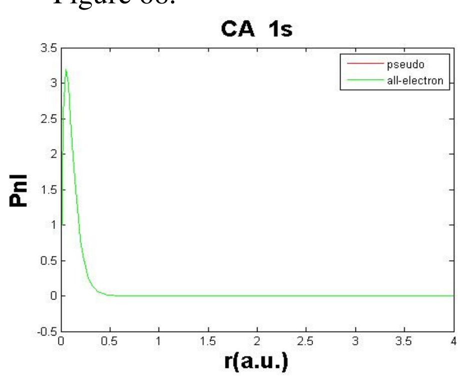


Figure 70.

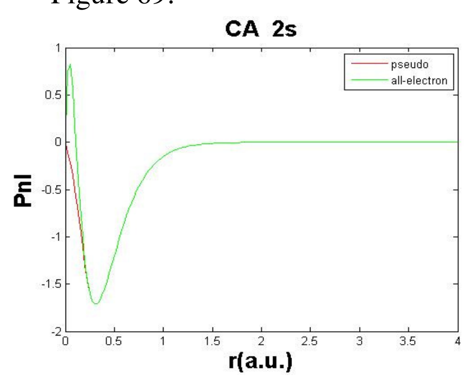


Figure 71.

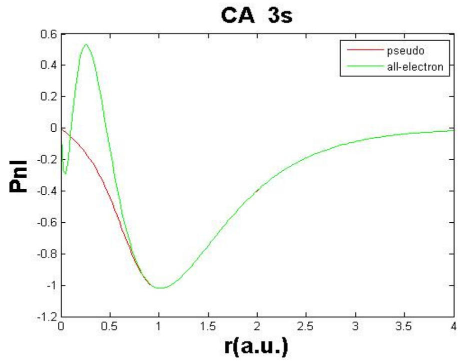


Figure 72.

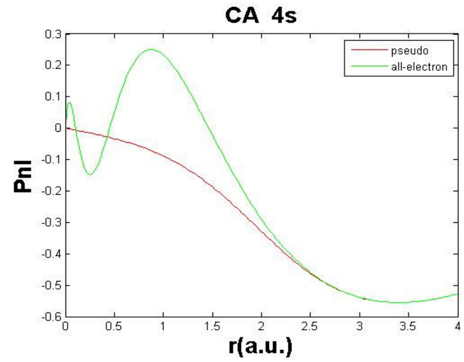


Figure 73.

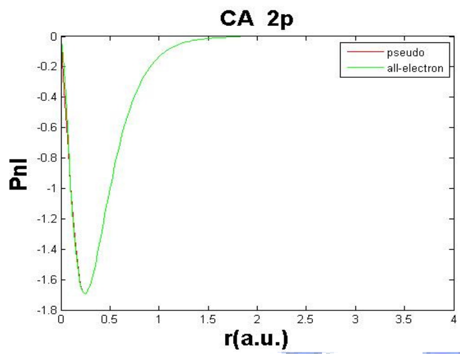


Figure 74.

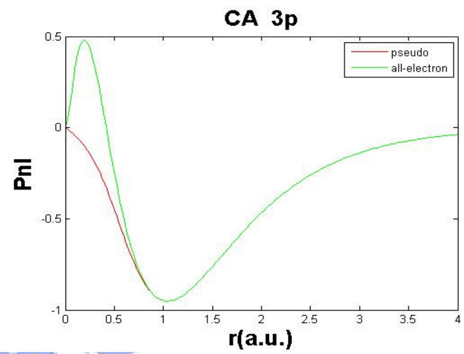


Figure 75.

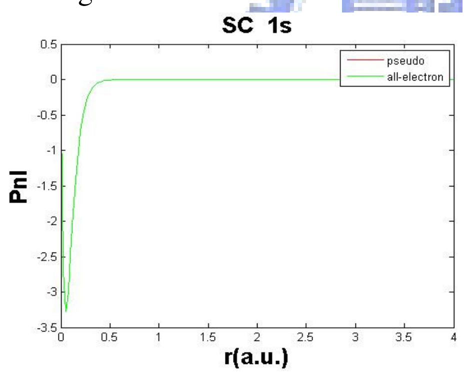


Figure 76.

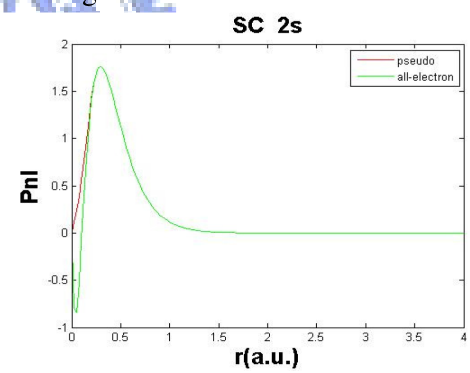


Figure 77.

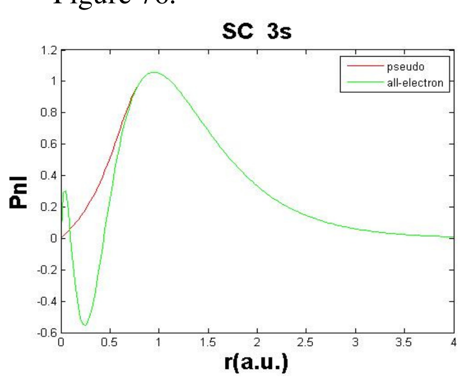


Figure 78.

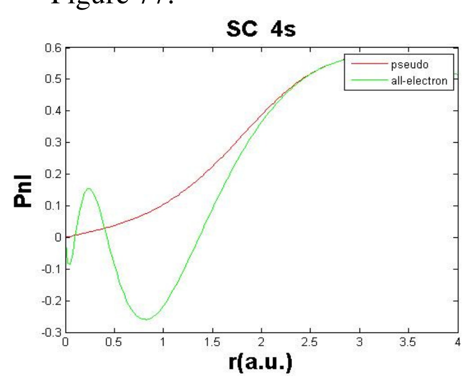


Figure 79.

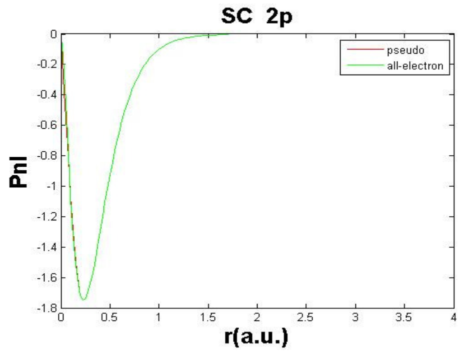


Figure 80.

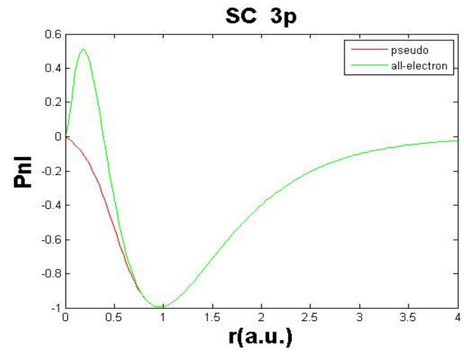


Figure 81.

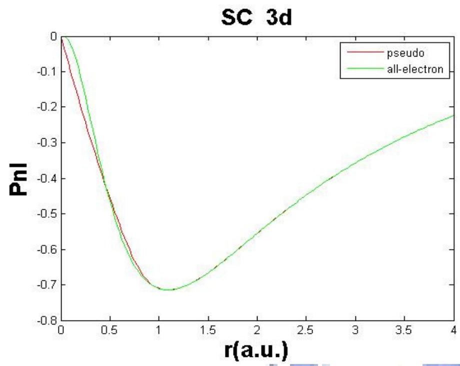


Figure 82.

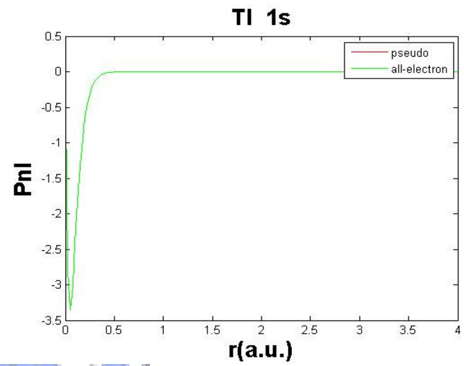


Figure 83.

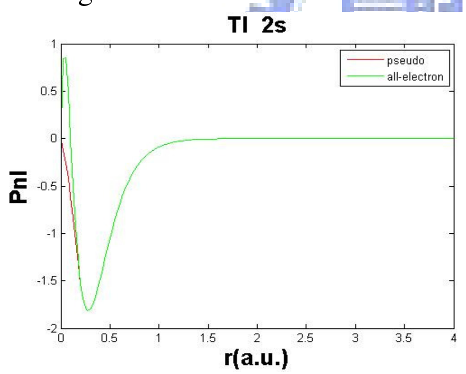


Figure 84.

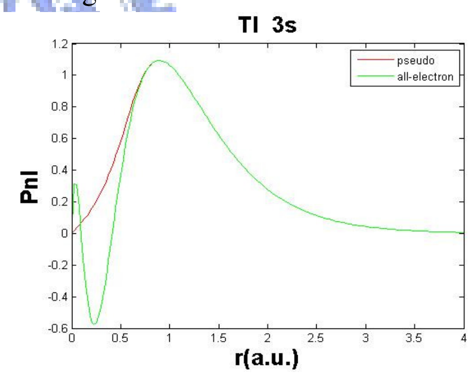


Figure 85.

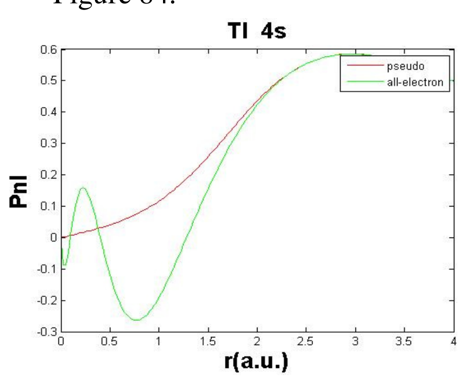


Figure 86.

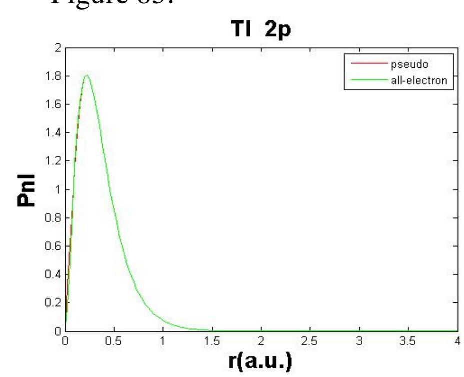


Figure 87.

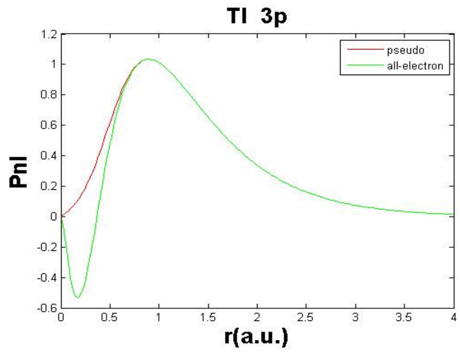


Figure 88.

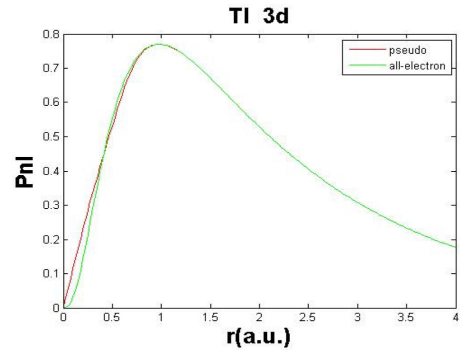


Figure 89.

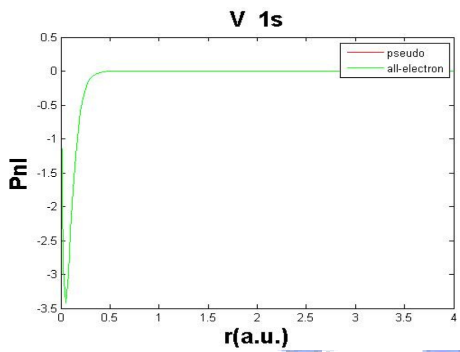


Figure 90.

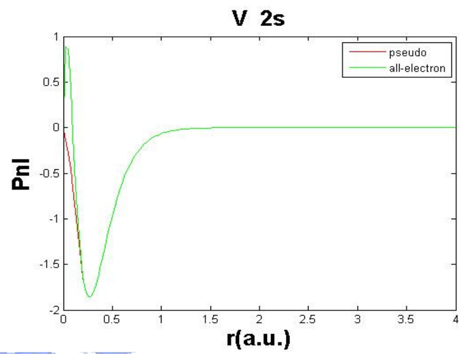


Figure 91.

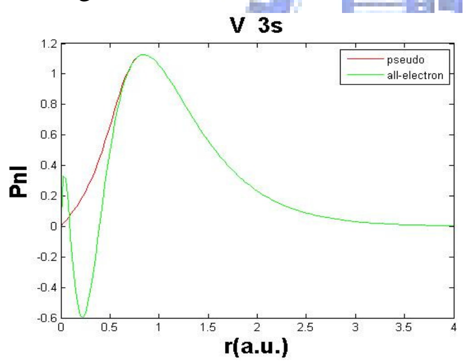


Figure 92.

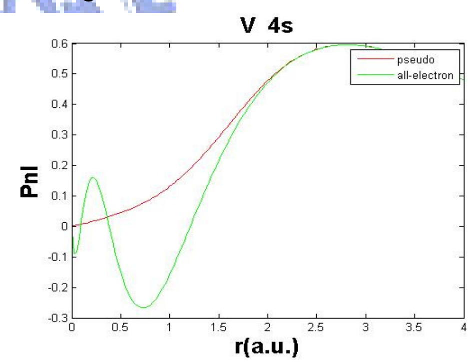


Figure 93.

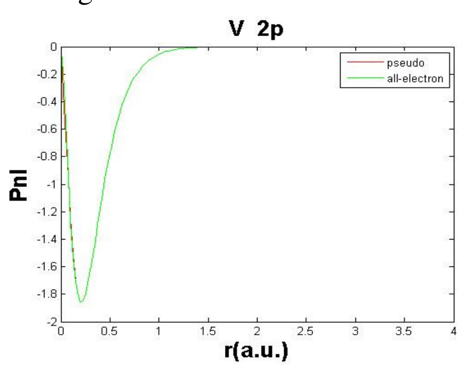


Figure 94.

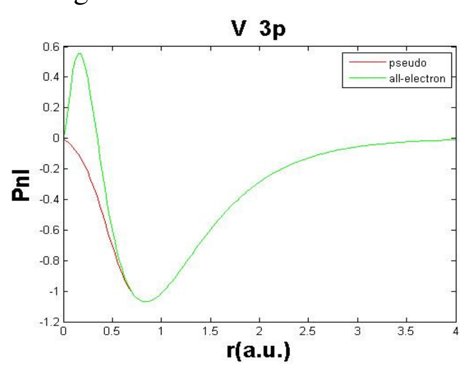


Figure 95.

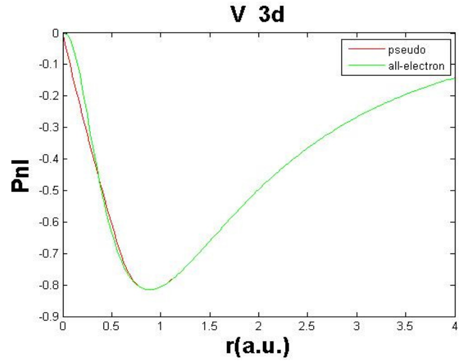


Figure 96.

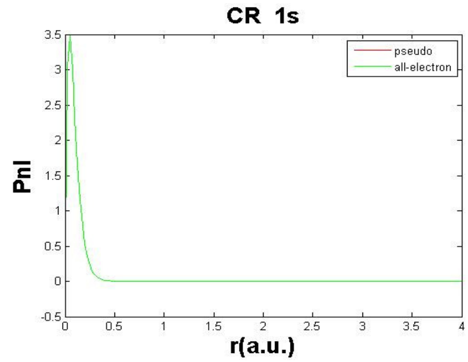


Figure 97.

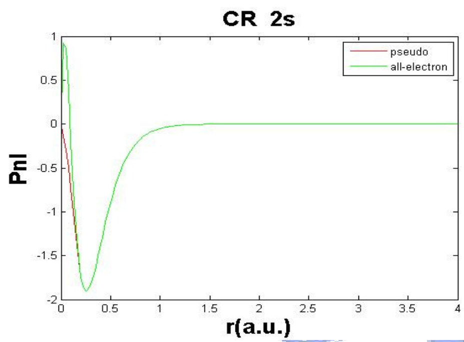


Figure 98.

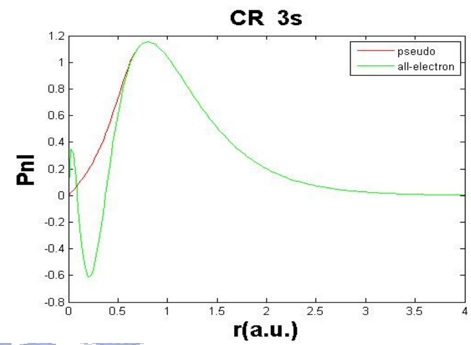


Figure 99.

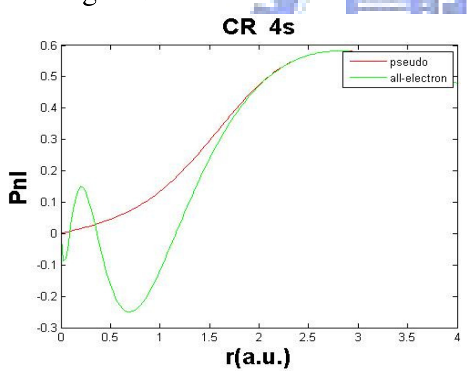


Figure 100.

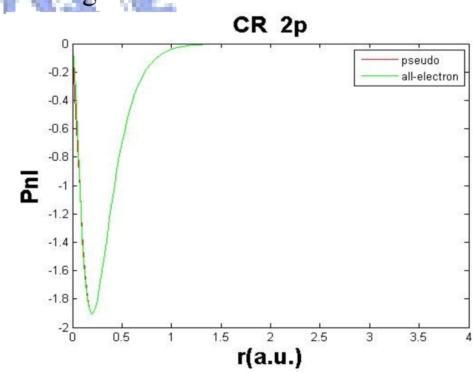


Figure 101.

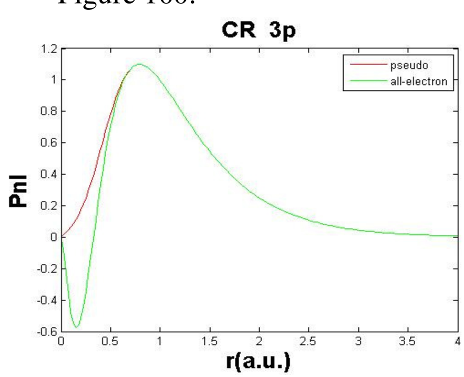


Figure 102.

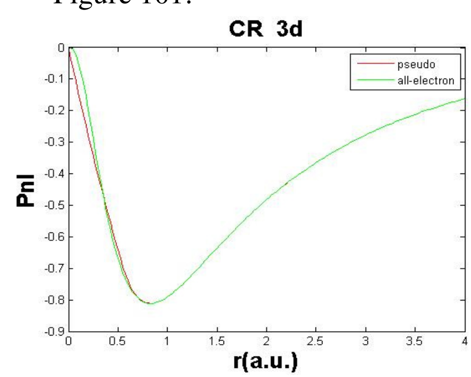


Figure 103.

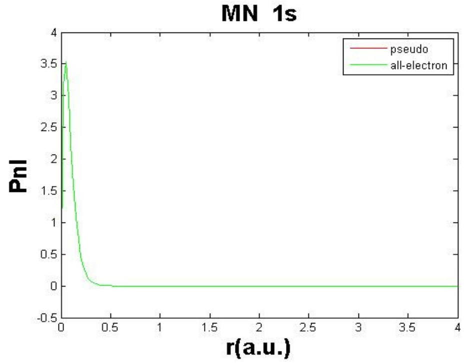


Figure 104.

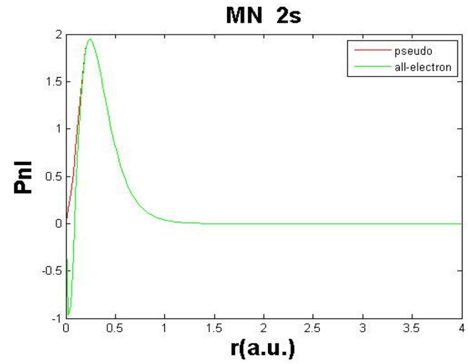


Figure 105.

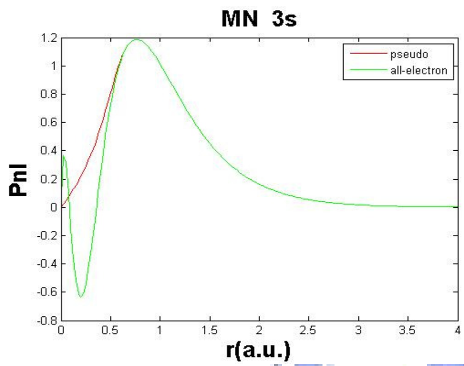


Figure 106.

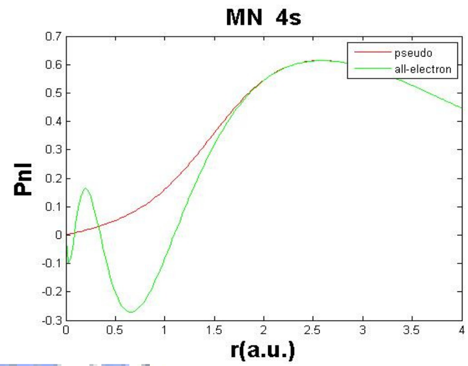


Figure 107.

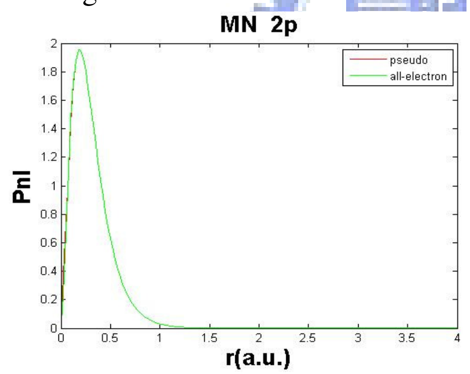


Figure 108.

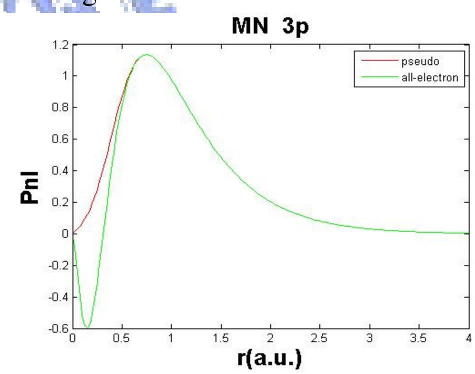


Figure 109.

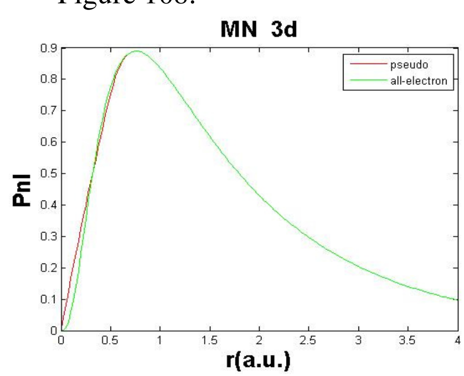


Figure 110.

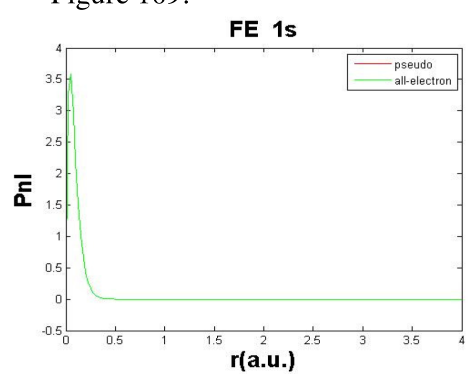


Figure 111.

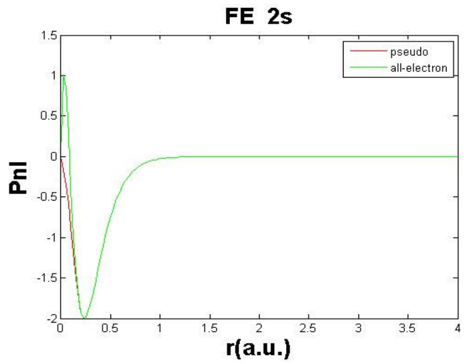


Figure 112.

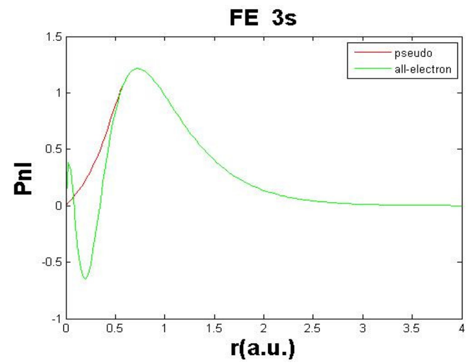


Figure 113.

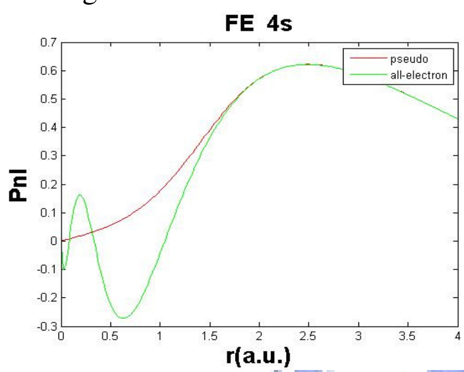


Figure 114.

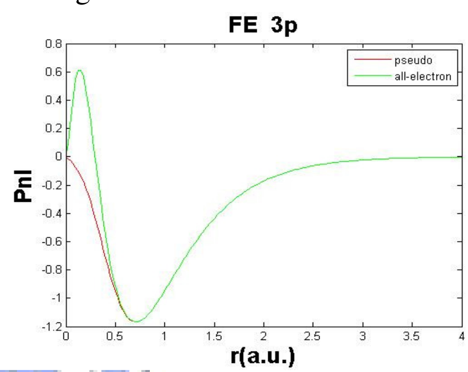


Figure 115.

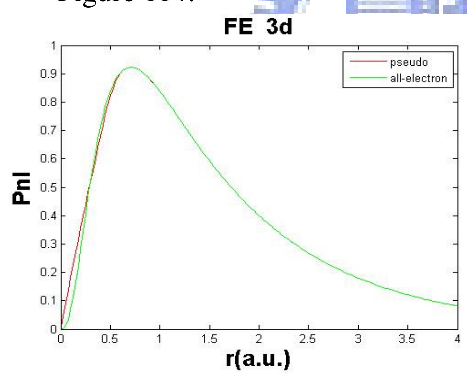


Figure 116.

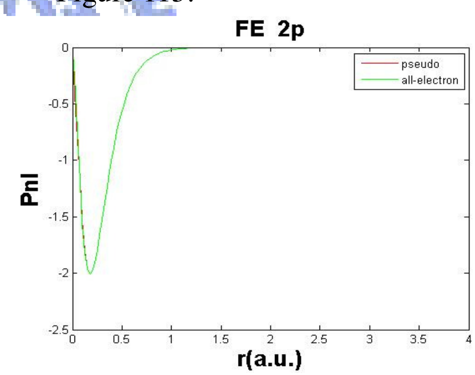


Figure 117.

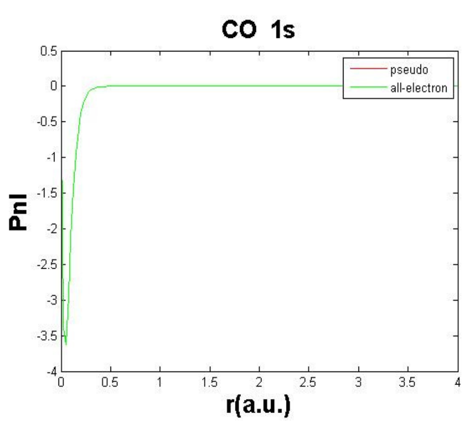


Figure 118.

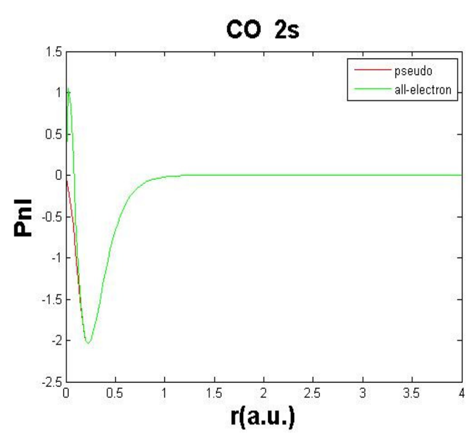


Figure 119.

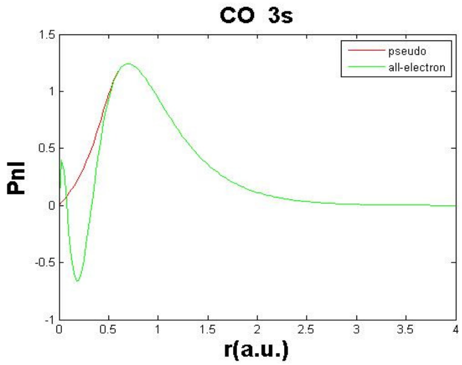


Figure 120.

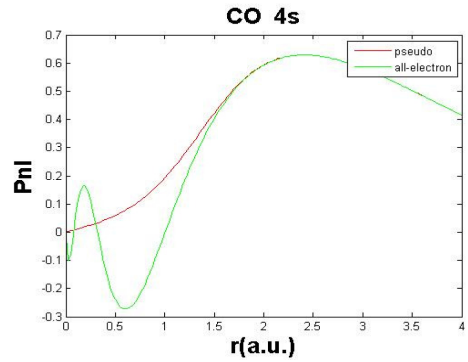


Figure 121.

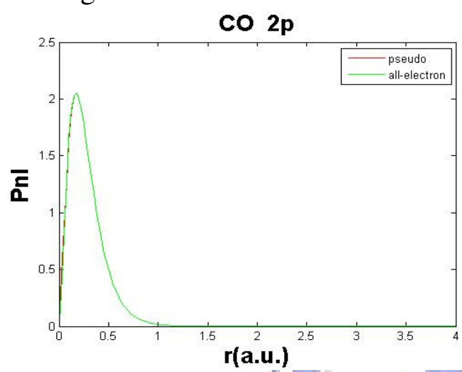


Figure 122.

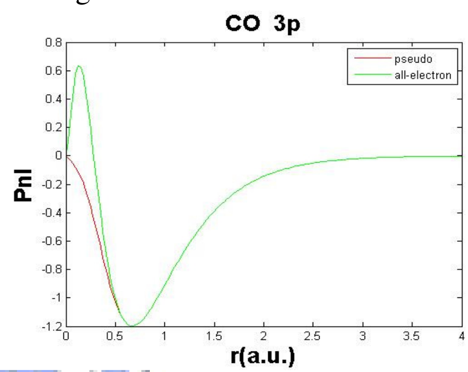


Figure 123.

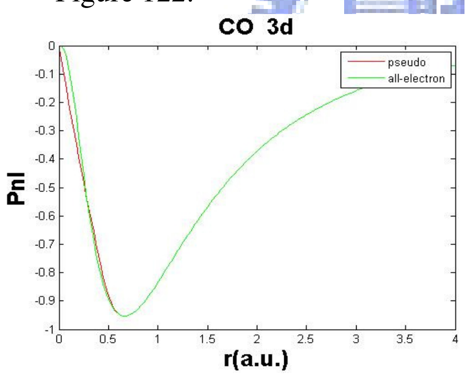


Figure 124.

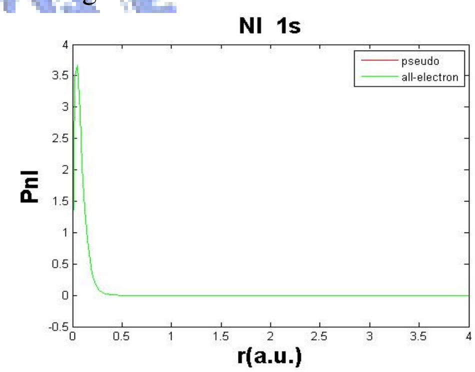


Figure 125.

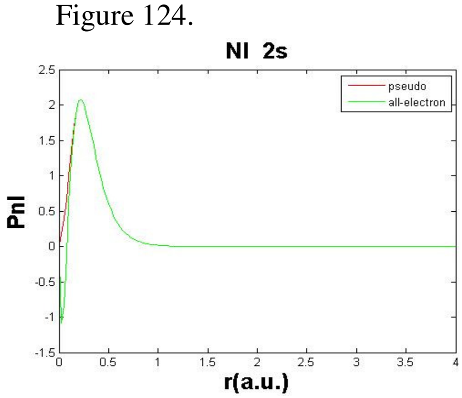


Figure 126.

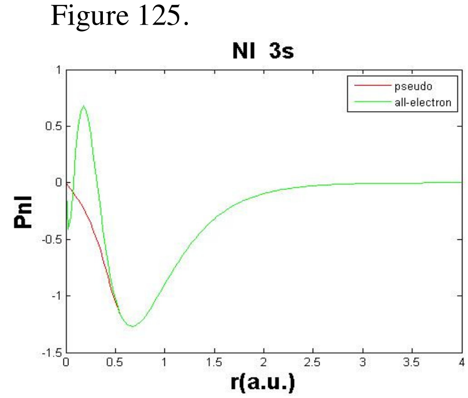


Figure 127.

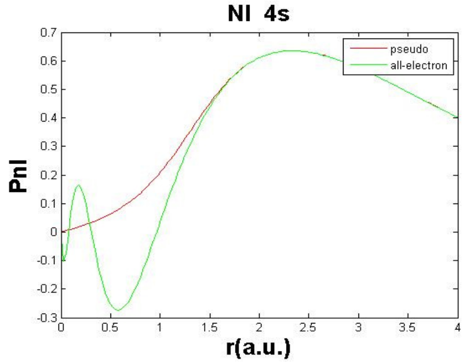


Figure 128.

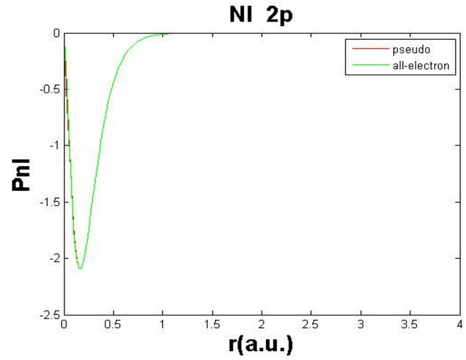


Figure 129.

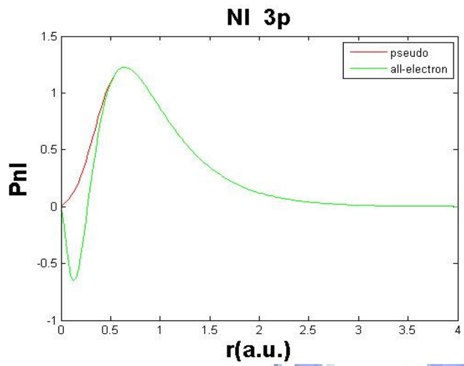


Figure 130.

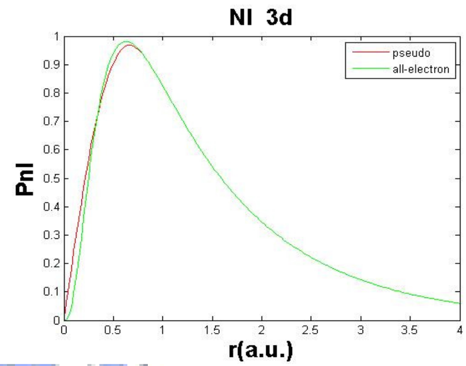


Figure 131.

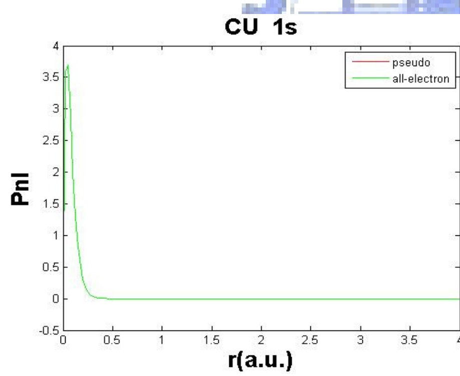


Figure 132.

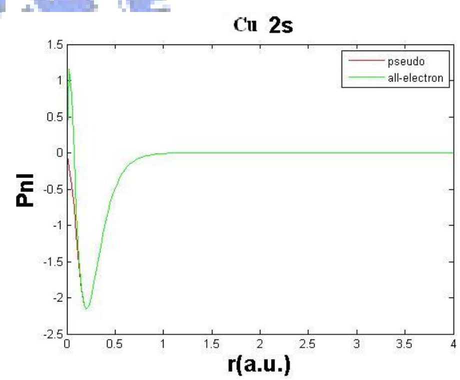


Figure 133.

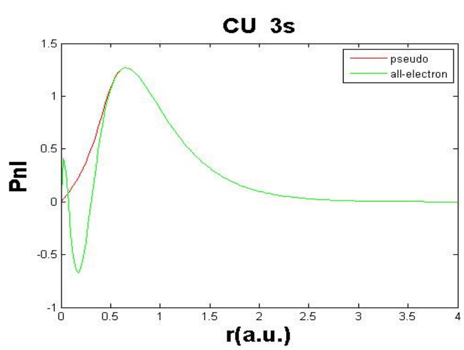


Figure 134.

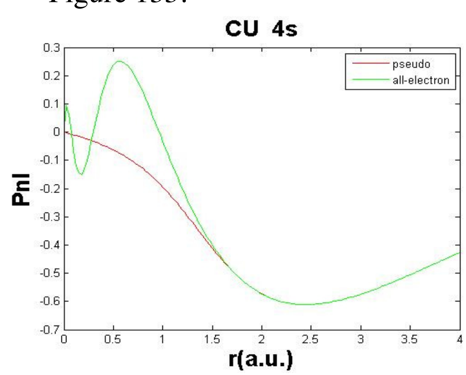


Figure 135.

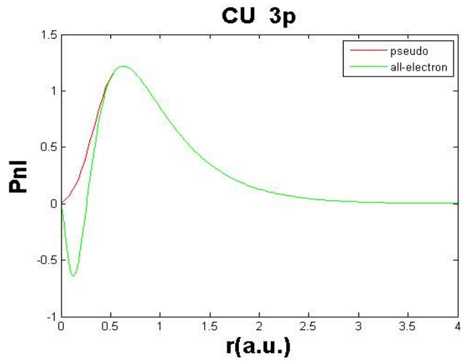


Figure 136.

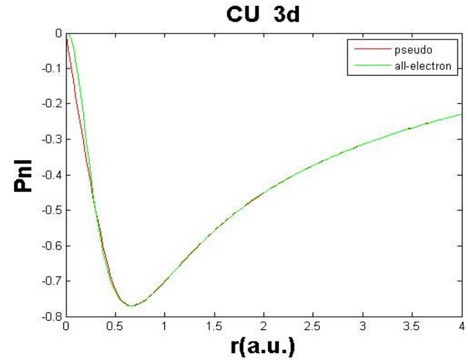


Figure 137.

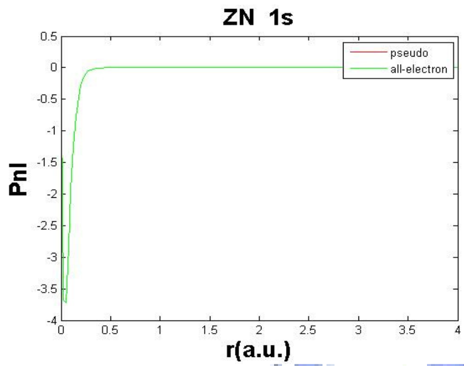


Figure 138.

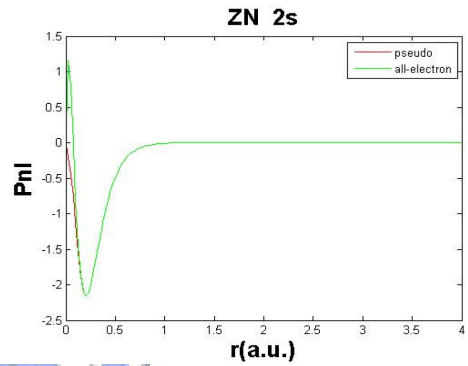


Figure 139.

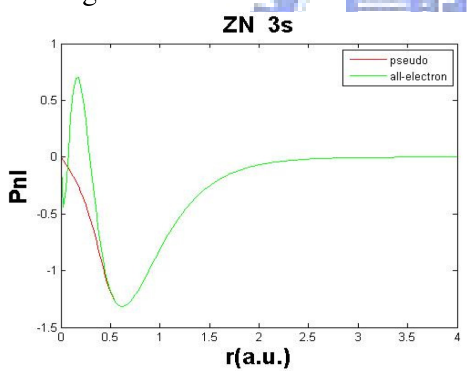


Figure 140.

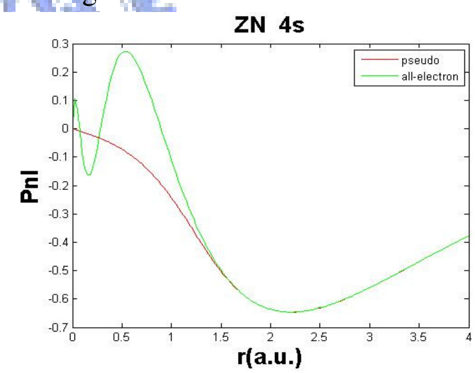


Figure 141.

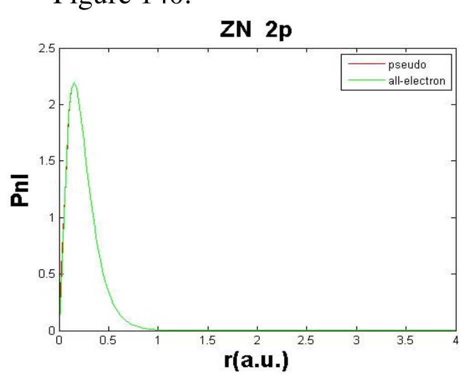


Figure 142.

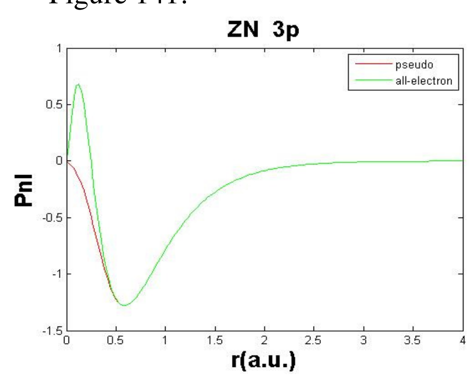


Figure 143.

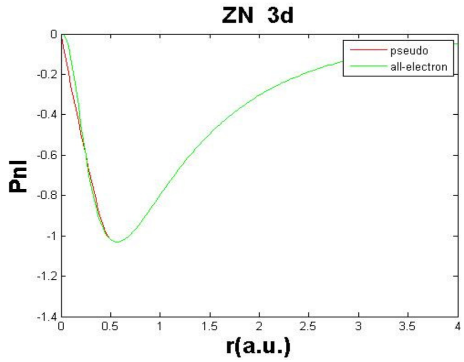


Figure 144.

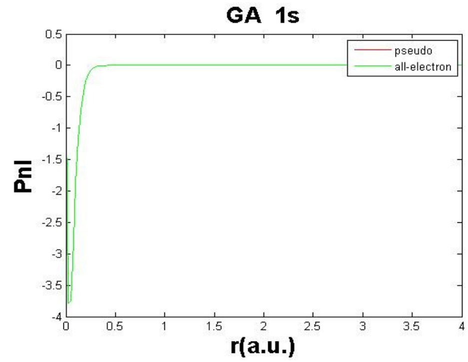


Figure 145.

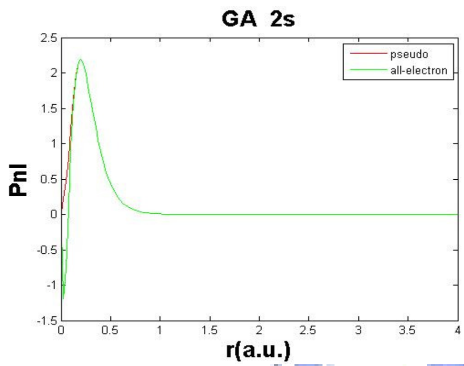


Figure 146.

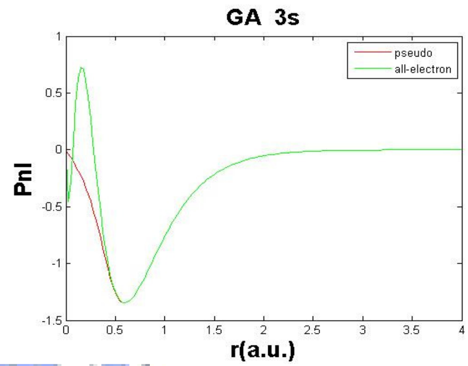


Figure 147.

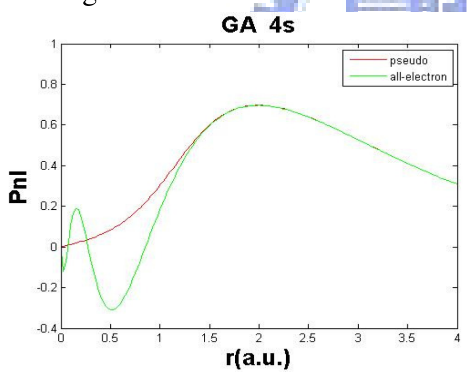


Figure 148.

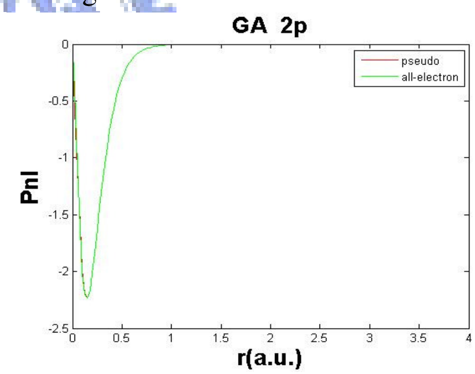


Figure 149.

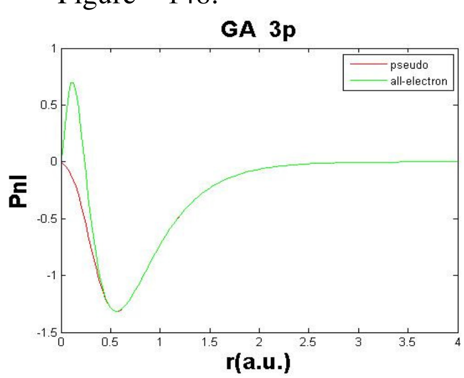


Figure 150.

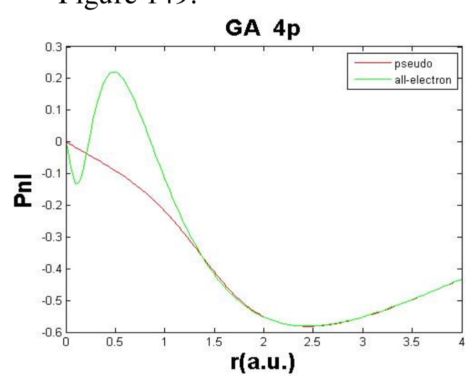


Figure 151.

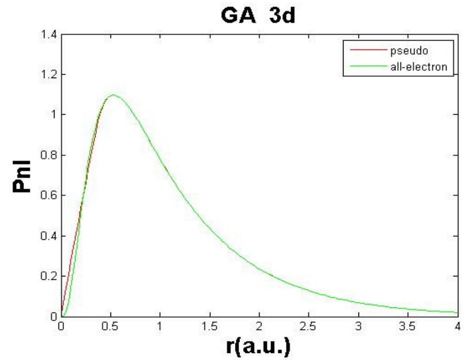


Figure 152.

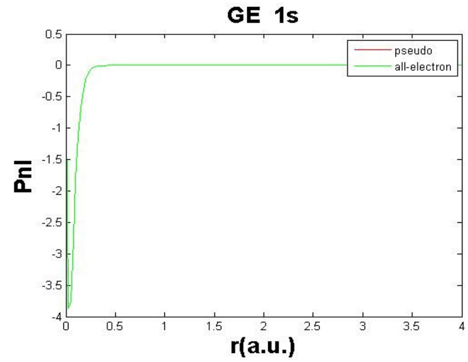


Figure 153.

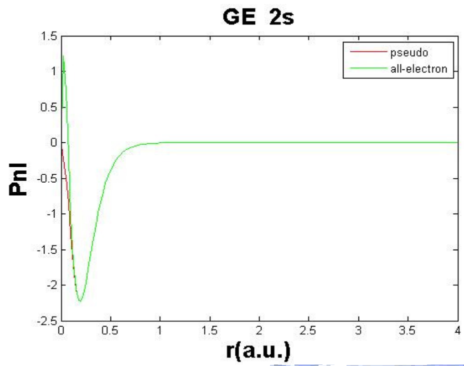


Figure 154.

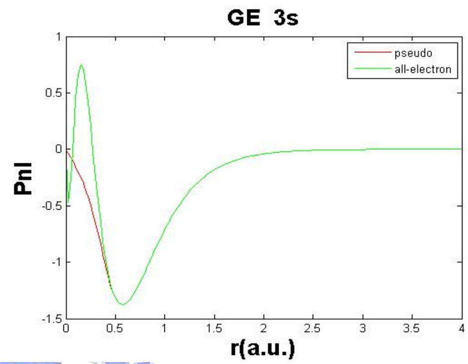


Figure 155.

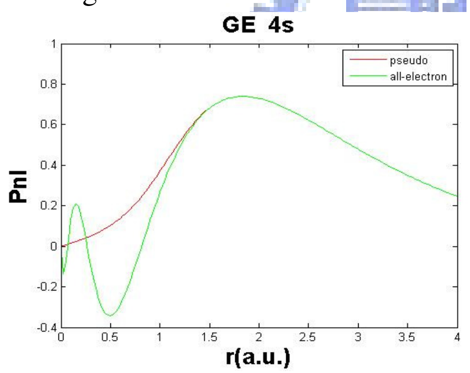


Figure 156.

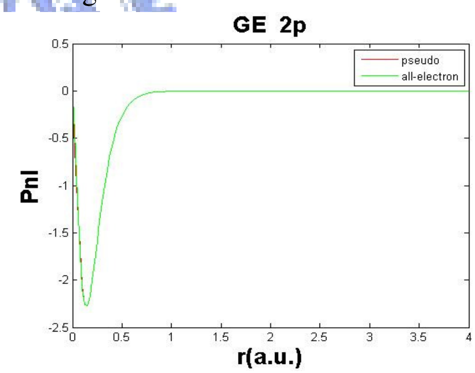


Figure 157.

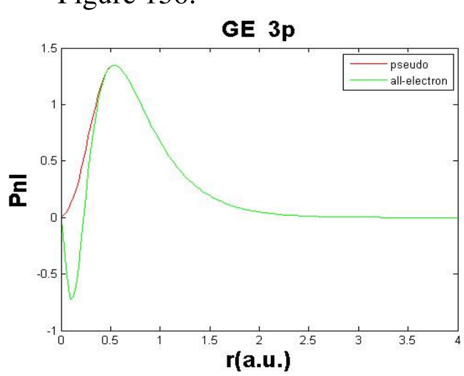


Figure 158.

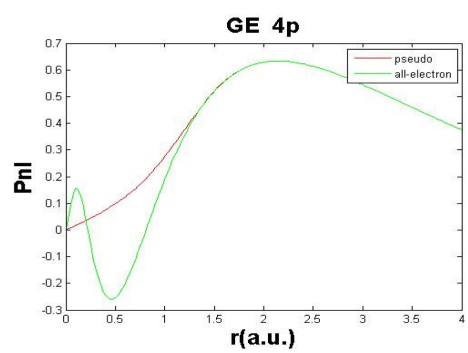


Figure 159.

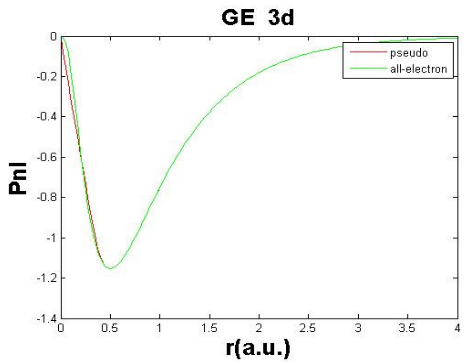


Figure 160.

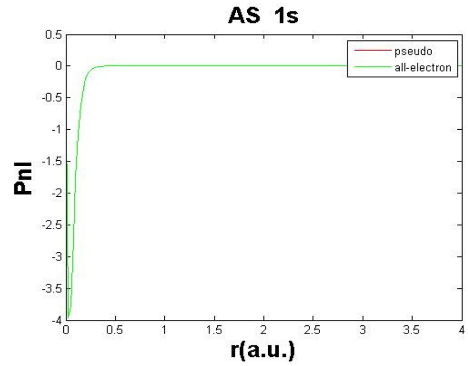


Figure 161.

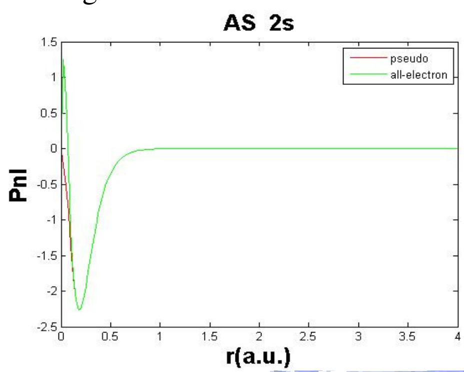


Figure 162.

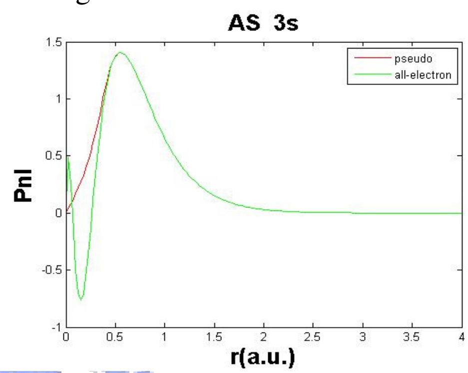


Figure 163.

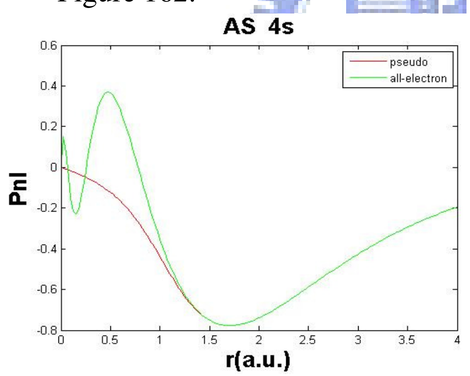


Figure 164.

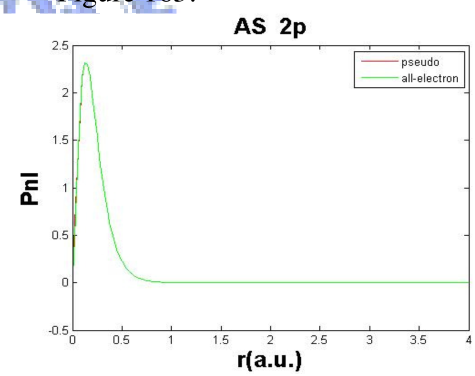


Figure 165.

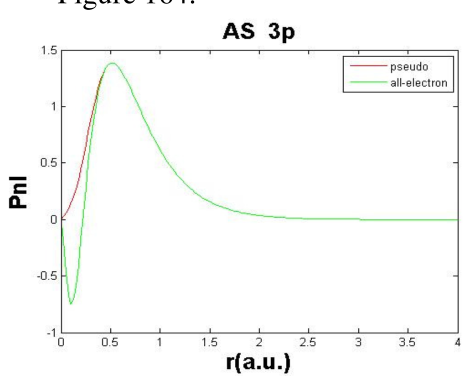


Figure 166.

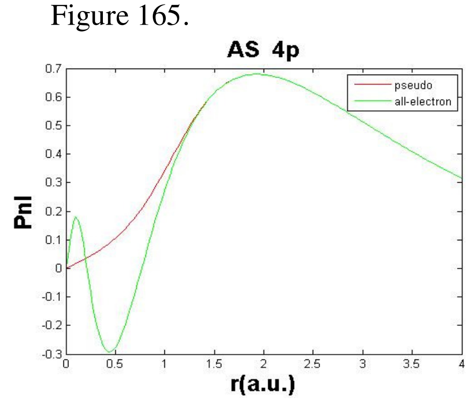


Figure 167.

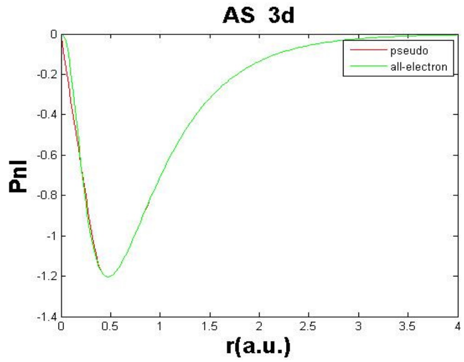


Figure 168.

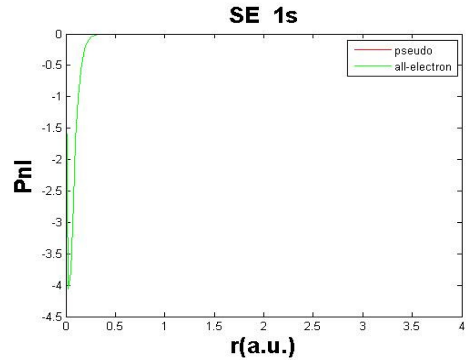


Figure 169.

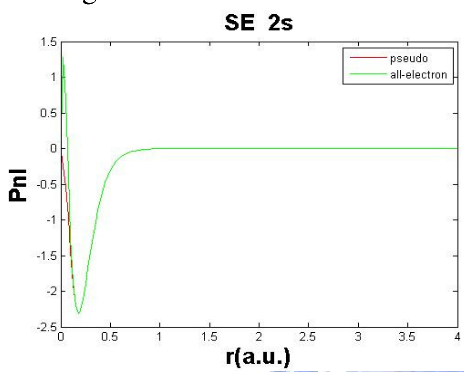


Figure 170.

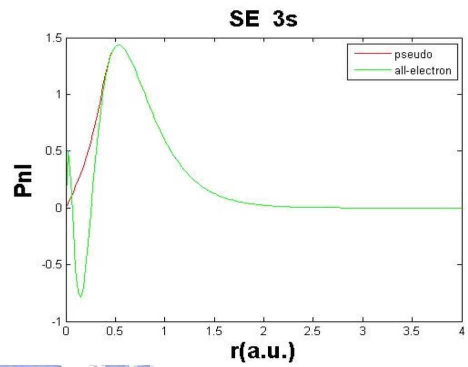


Figure 171.

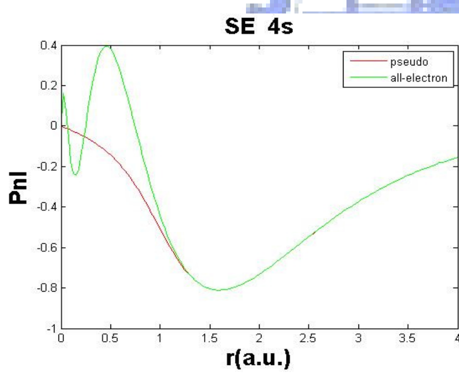


Figure 172.

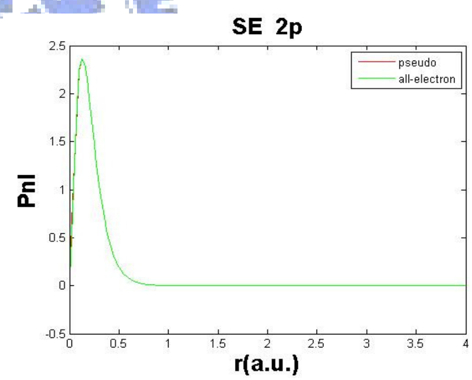


Figure 173.

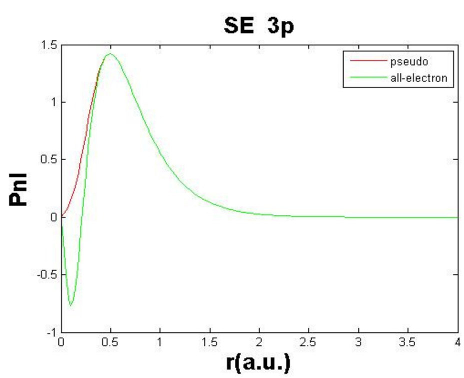


Figure 174.

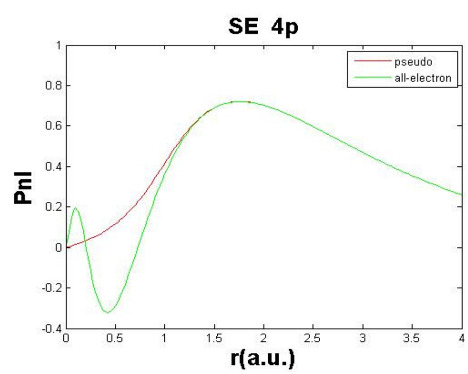


Figure 175.

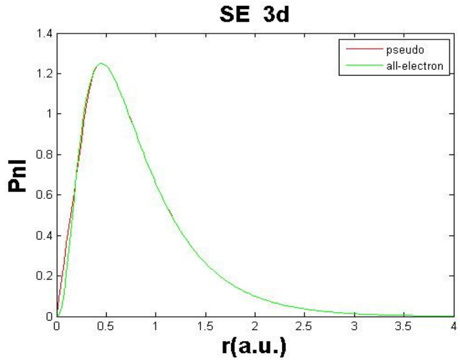


Figure 176.

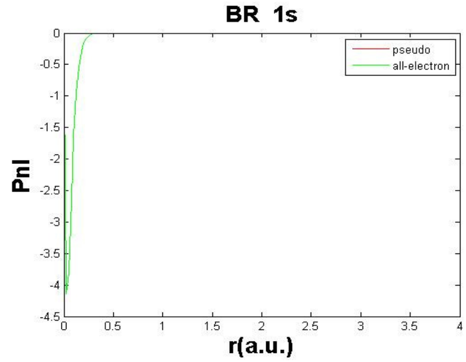


Figure 177.

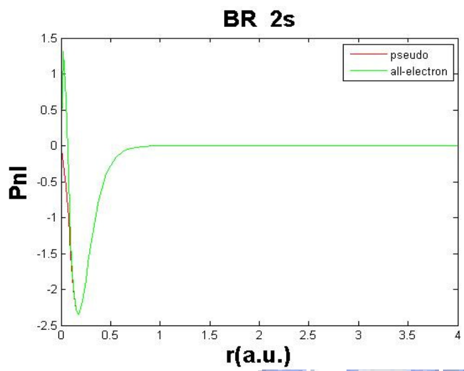


Figure 178.

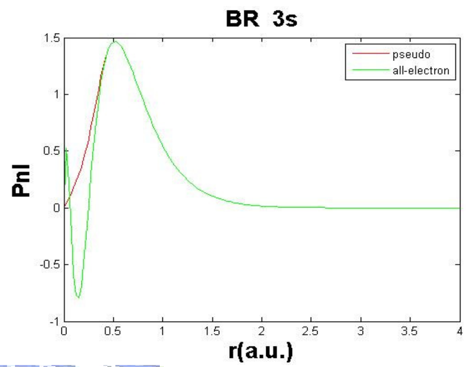


Figure 179.

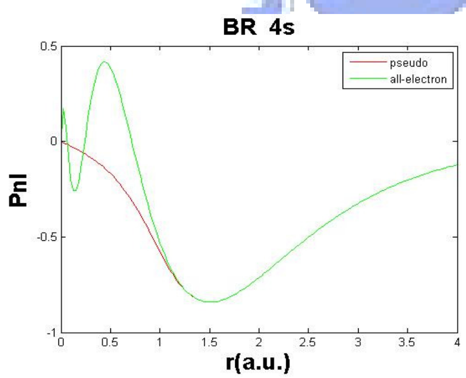


Figure 180.

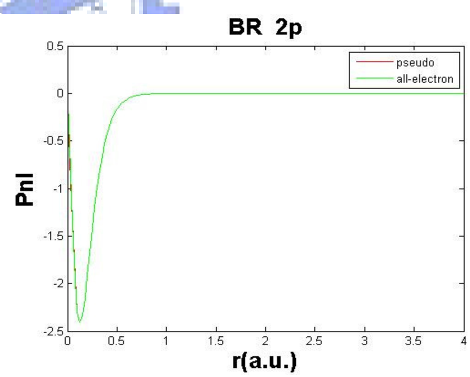


Figure 181.

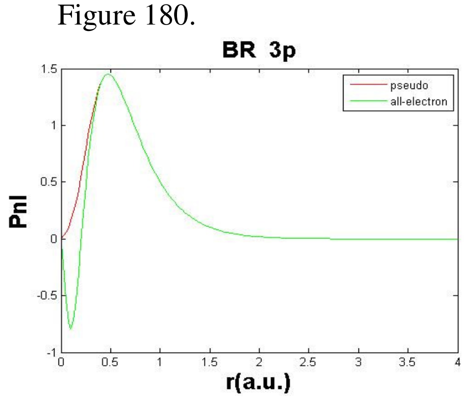


Figure 182.

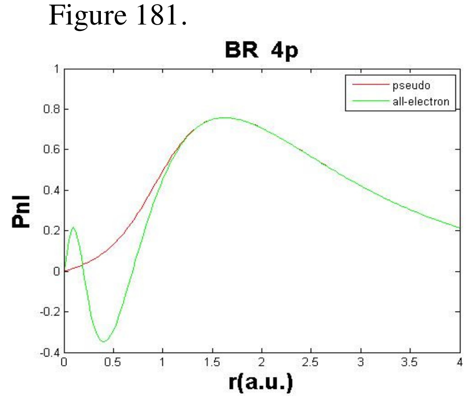


Figure 183.

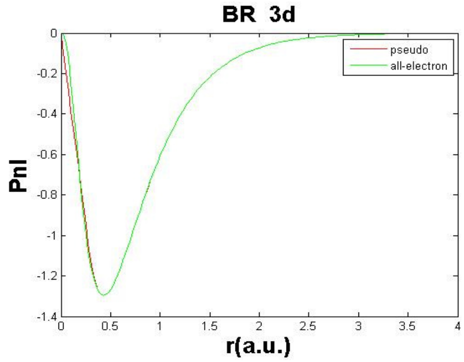


Figure 184.

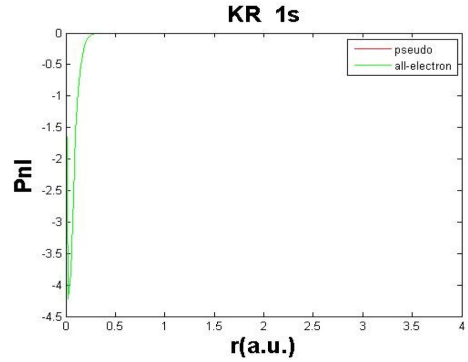


Figure 185.

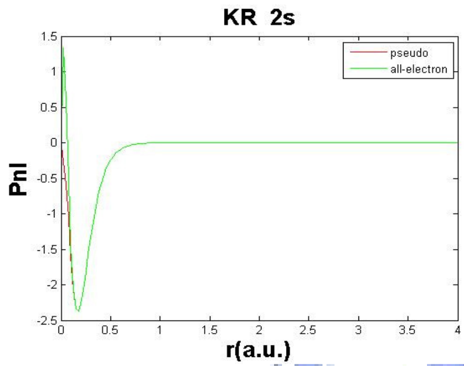


Figure 186.

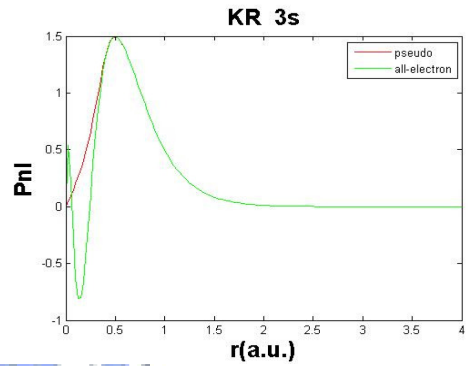


Figure 187.

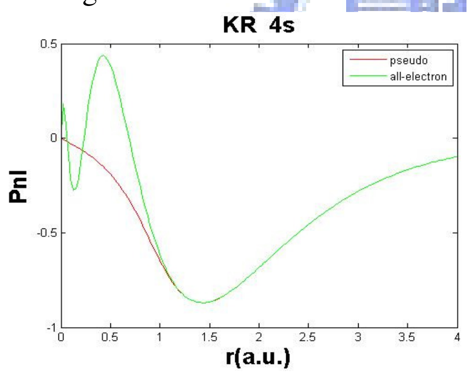


Figure 188.

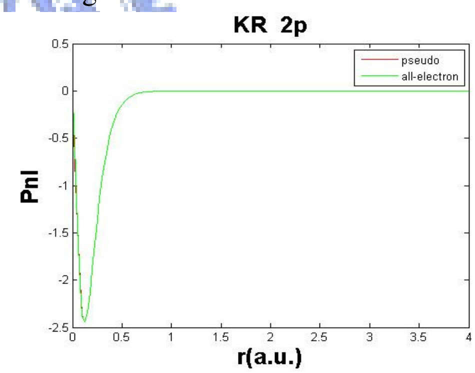


Figure 189.

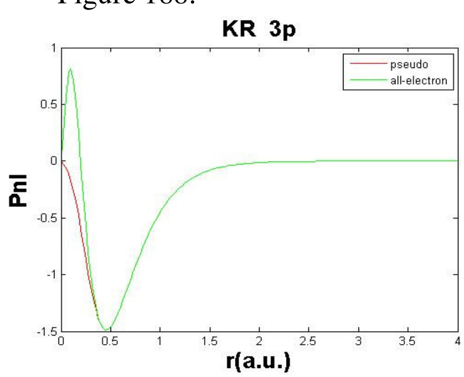


Figure 190.

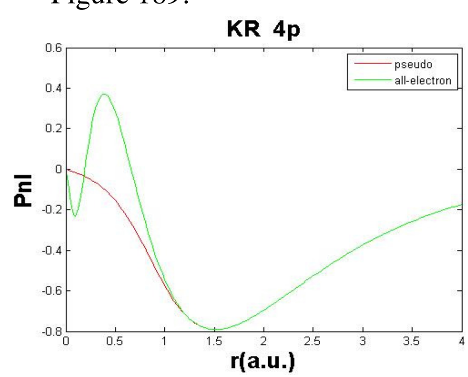


Figure 191.

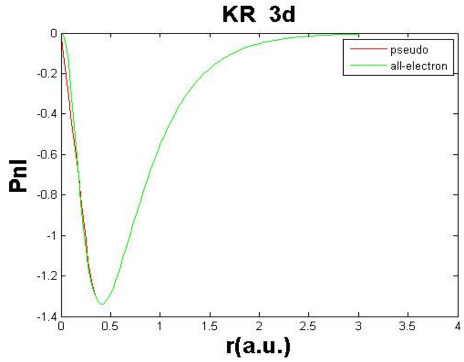


Figure 192.

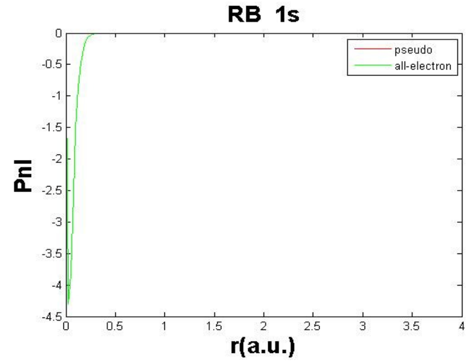


Figure 193.

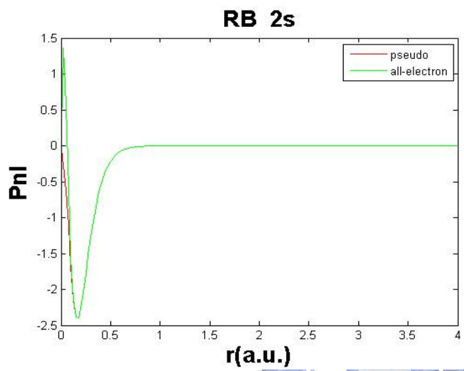


Figure 194.

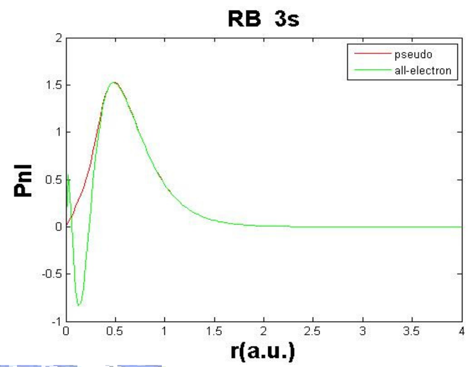


Figure 195.

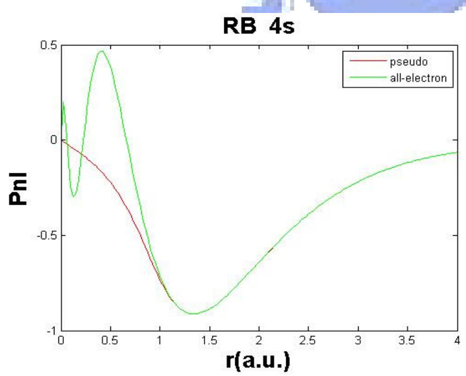


Figure 196.

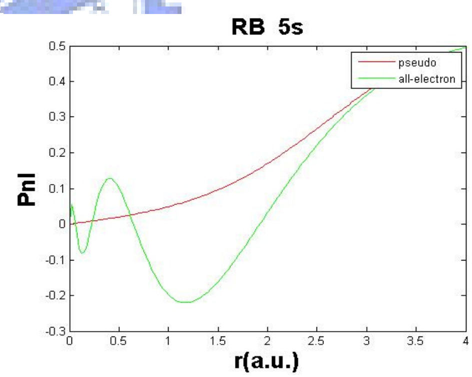


Figure 197.

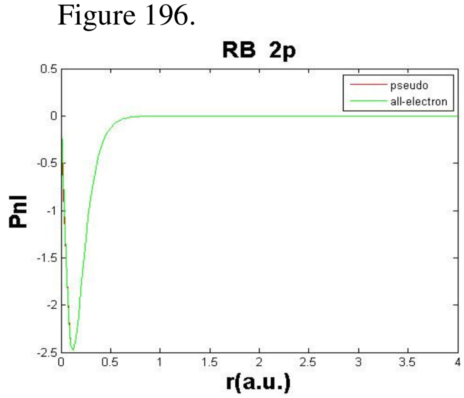


Figure 198.

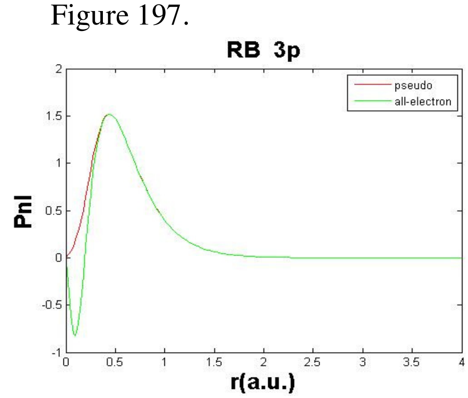


Figure 199.

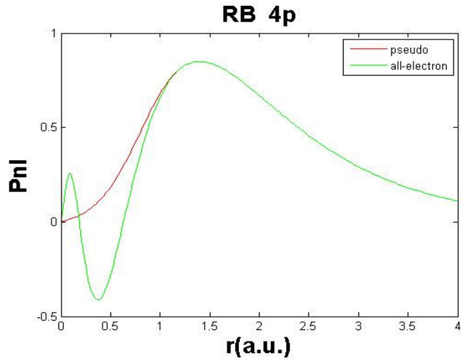


Figure 200.

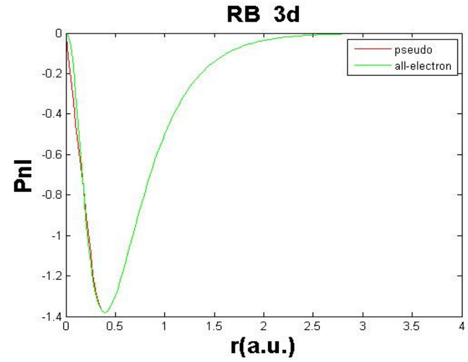


Figure 201.

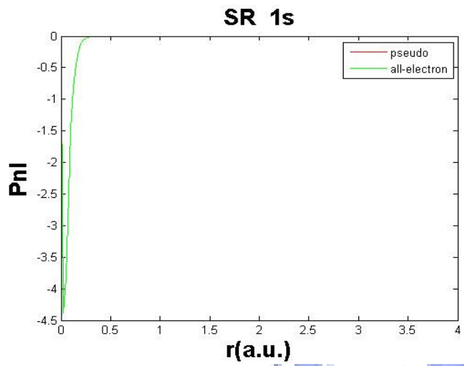


Figure 202.

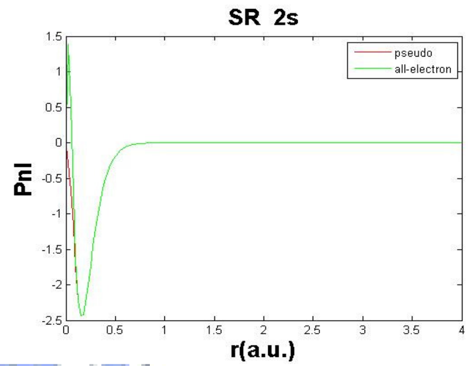


Figure 203.

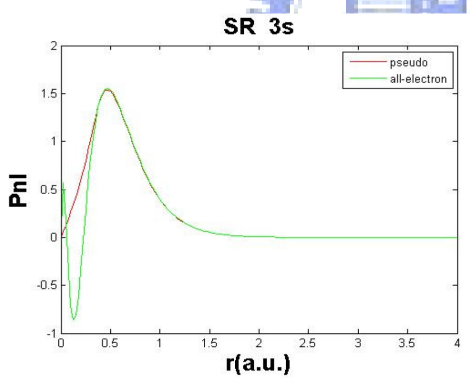


Figure 204.

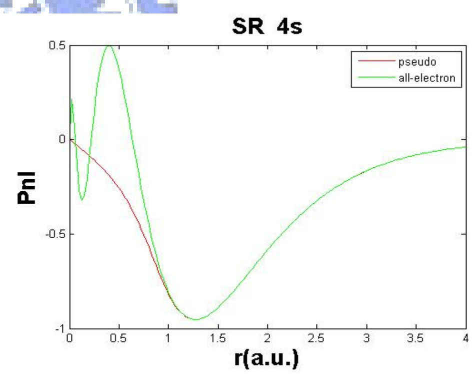


Figure 205.

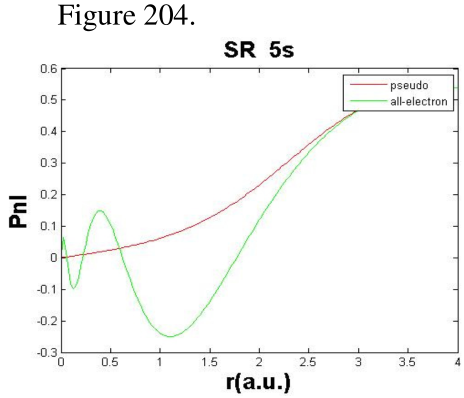


Figure 206.

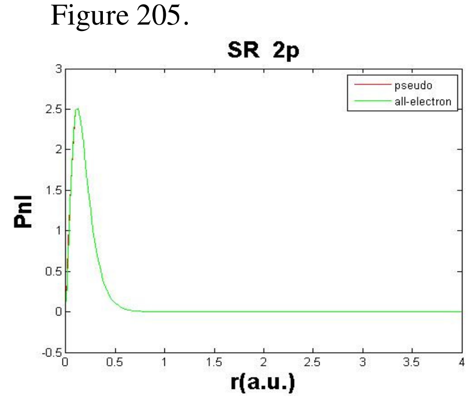


Figure 207.

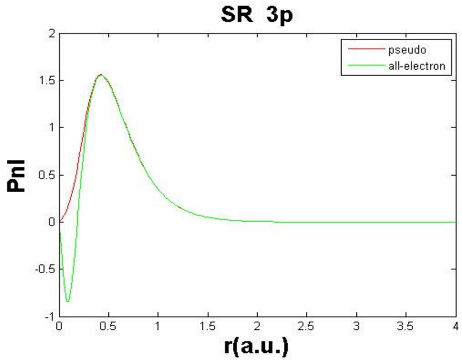


Figure 208.

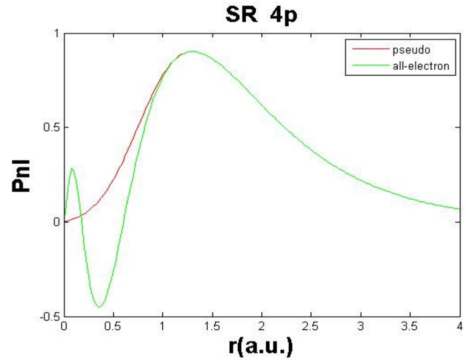


Figure 209.

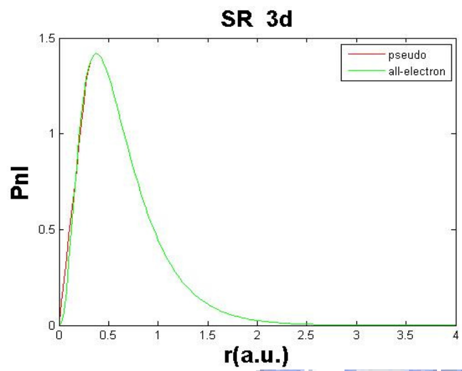


Figure 210.

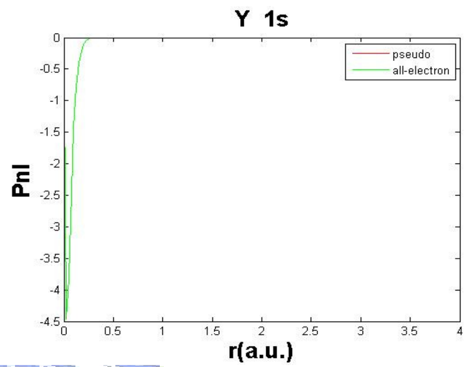


Figure 211.

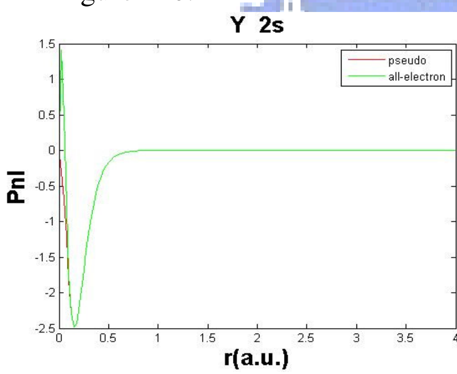


Figure 212.

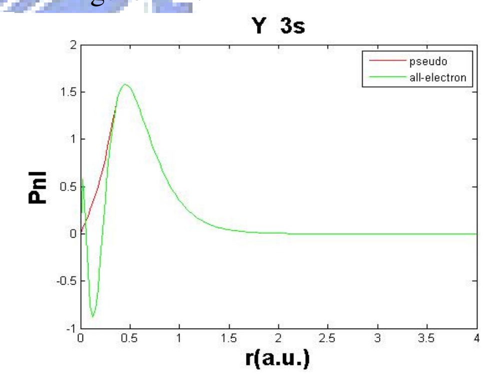


Figure 213.

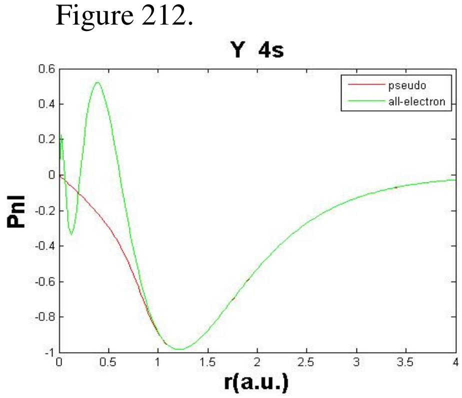


Figure 214.

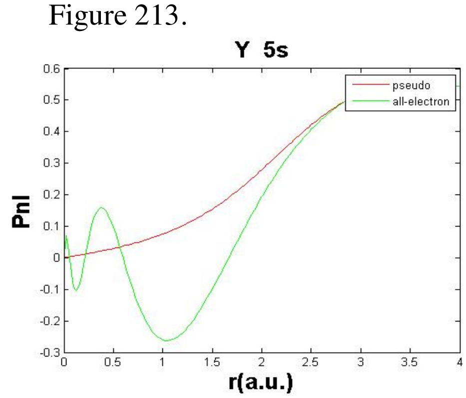


Figure 215.

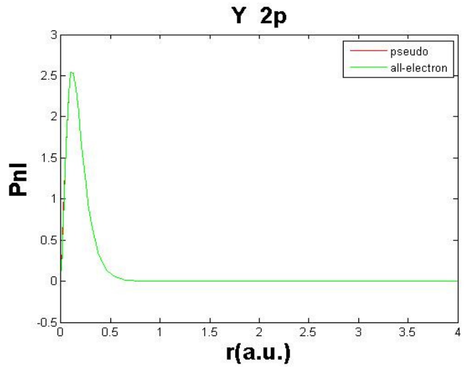


Figure 216.

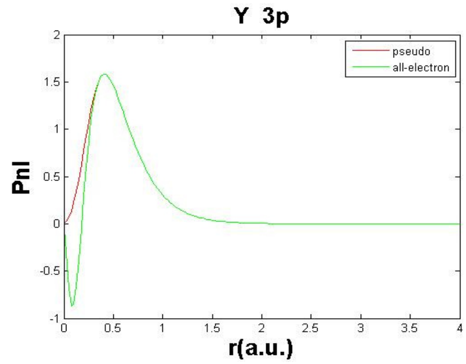


Figure 217.

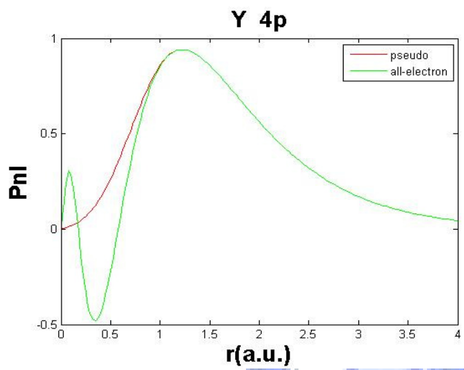


Figure 218.

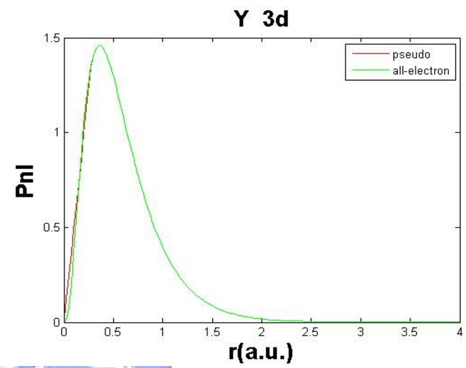


Figure 219.

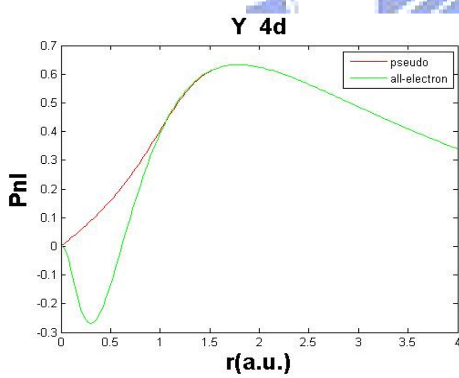


Figure 220.

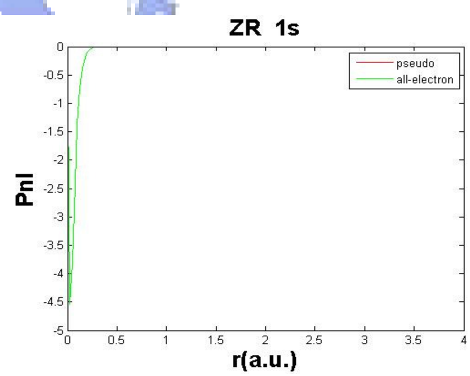


Figure 221.

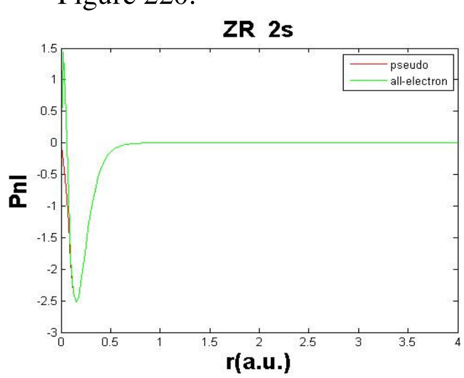


Figure 222.

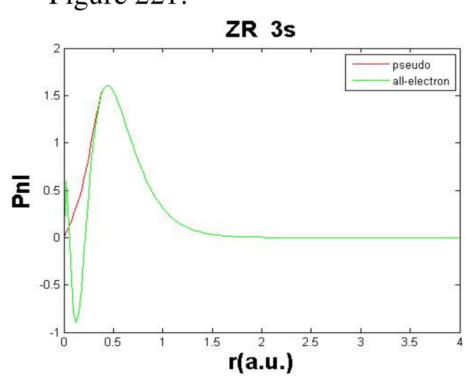


Figure 223.

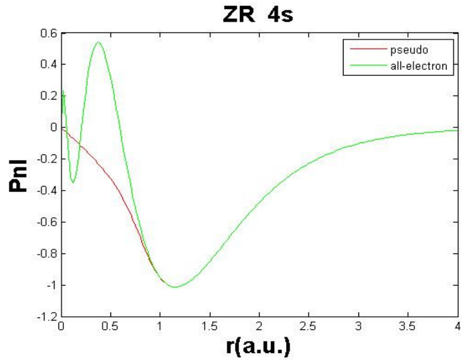


Figure 224.

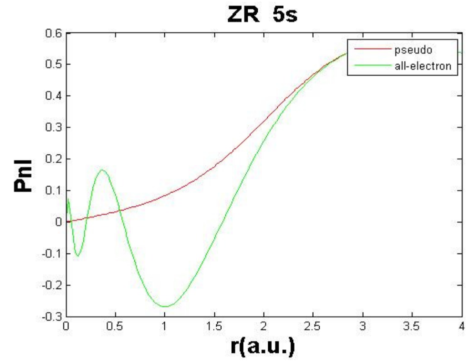


Figure 225.

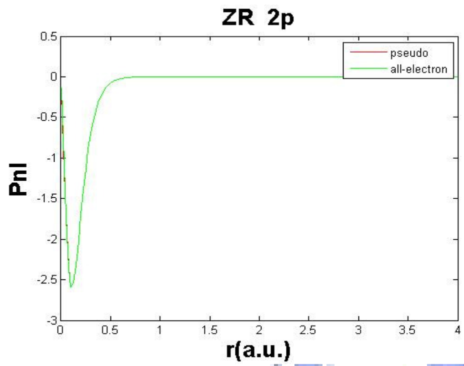


Figure 226.

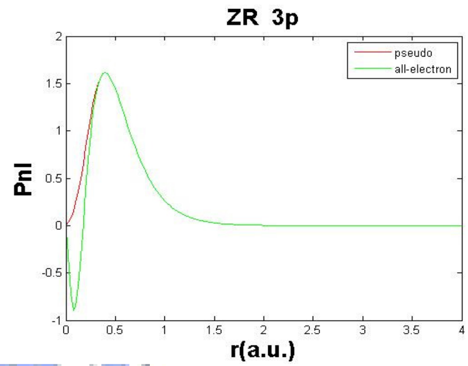


Figure 227.

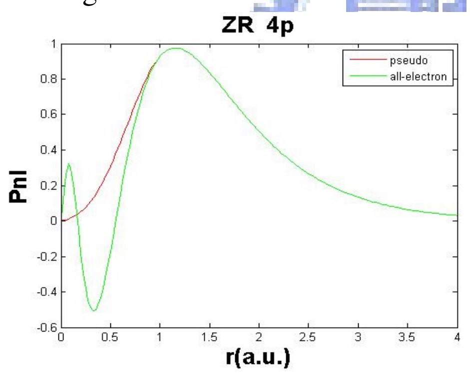


Figure 228.

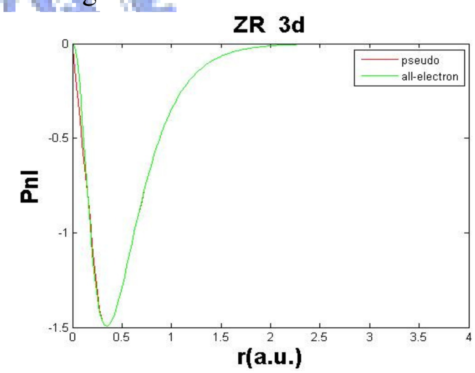


Figure 229.

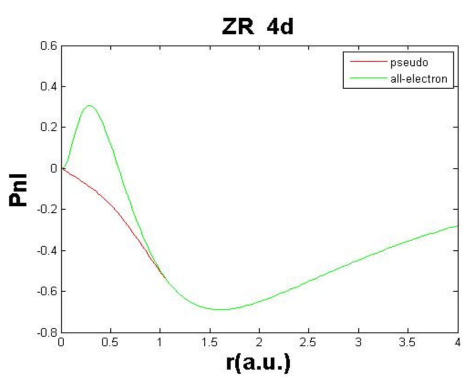


Figure 230.

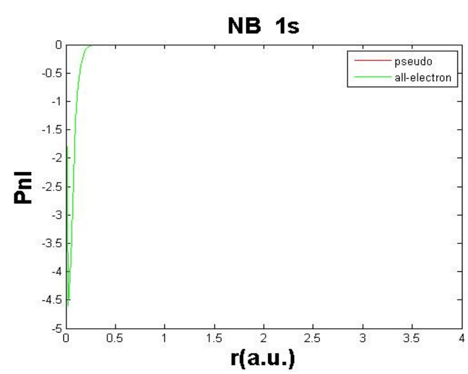


Figure 231.

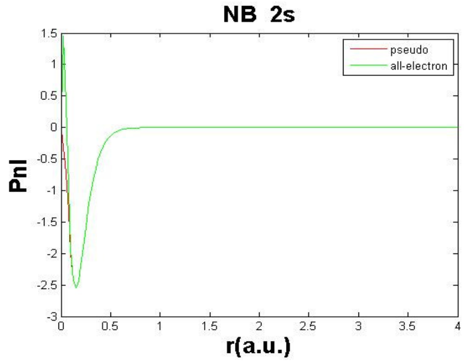


Figure 232.

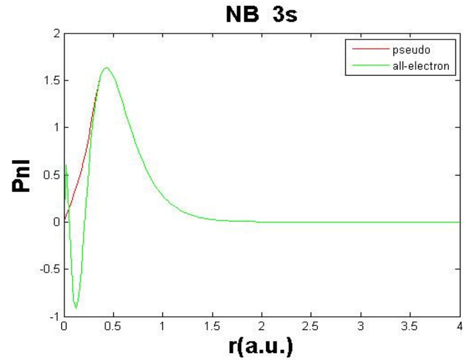


Figure 233.

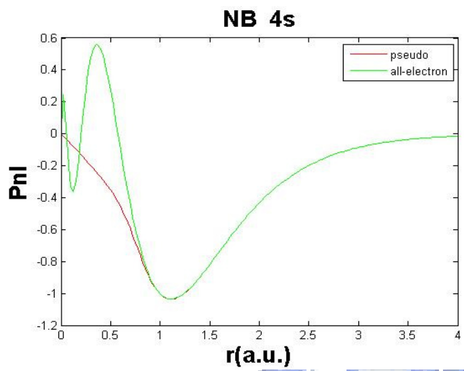


Figure 234.

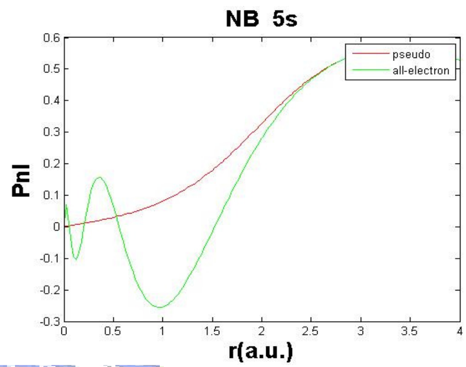


Figure 235.

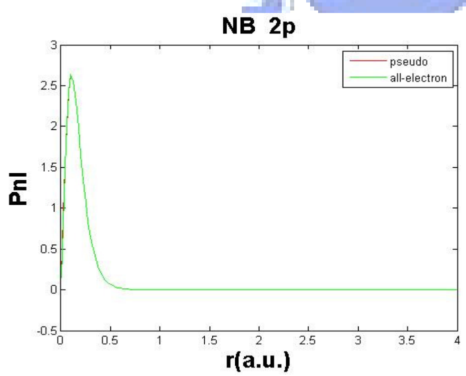


Figure 236.

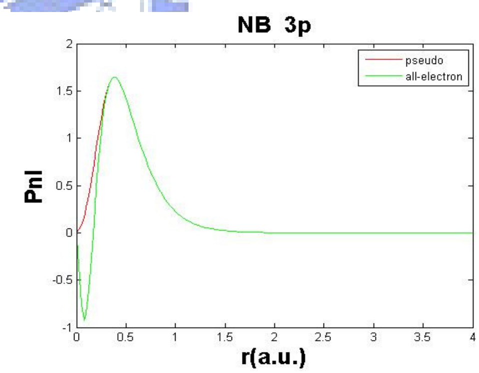


Figure 237.

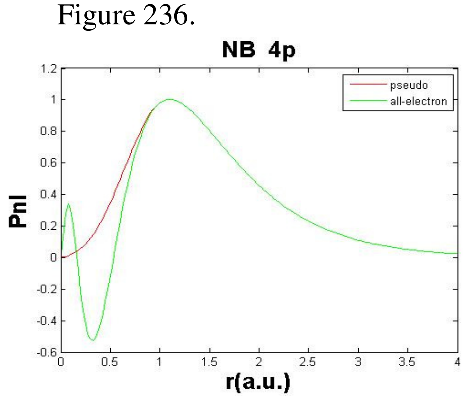


Figure 238.

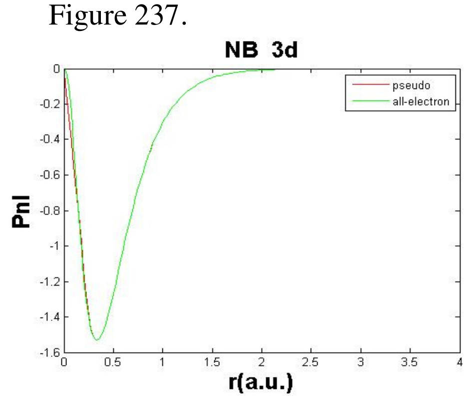


Figure 239.

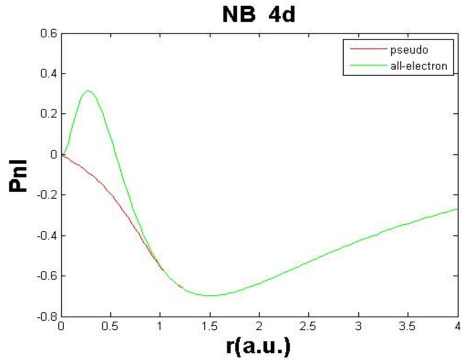


Figure 240.

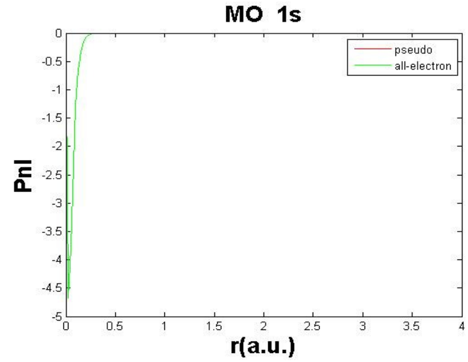


Figure 241.

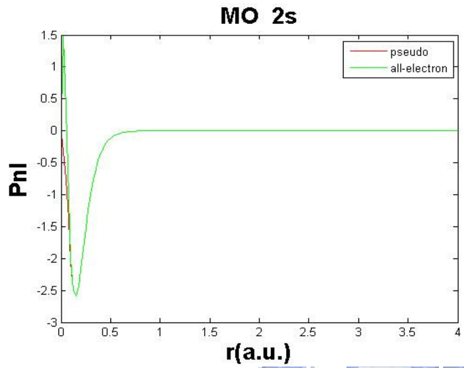


Figure 242.

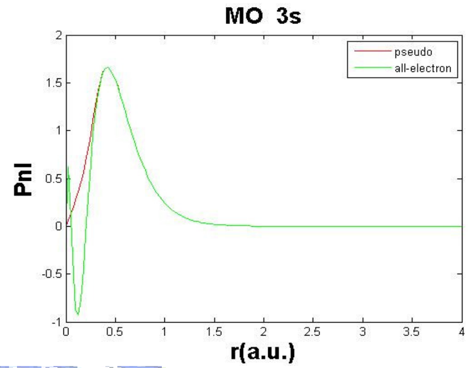


Figure 243.

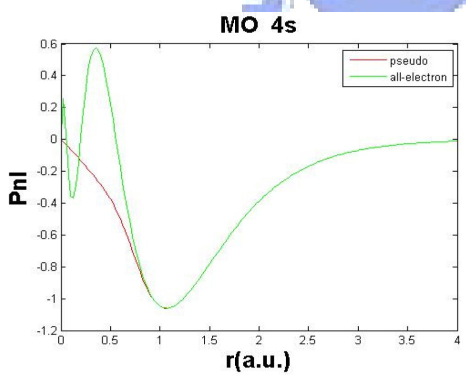


Figure 244.

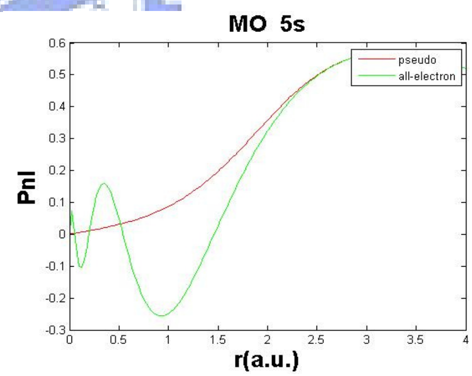


Figure 245.

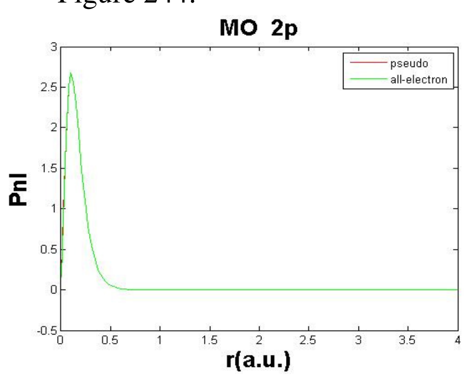


Figure 246.

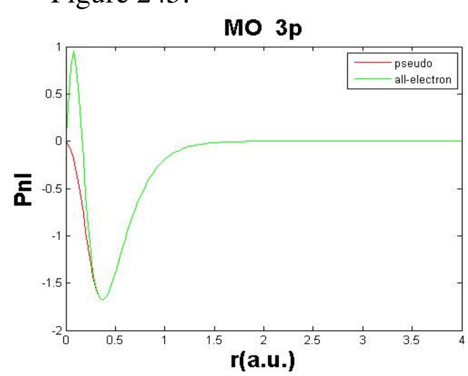


Figure 247.

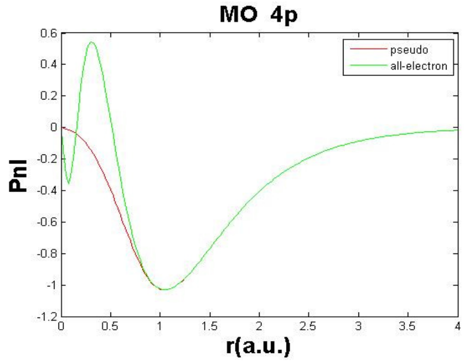


Figure 248.

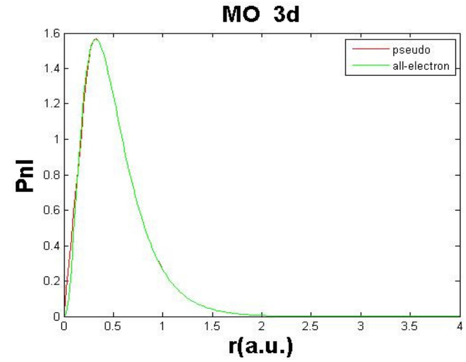


Figure 249.

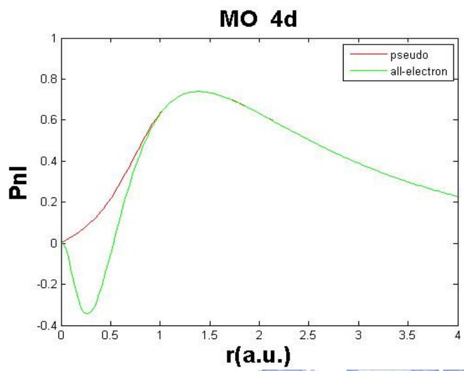


Figure 250.

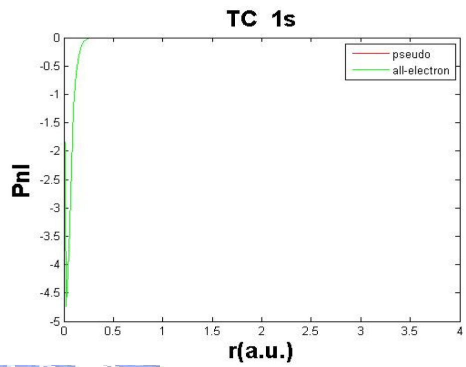


Figure 251.

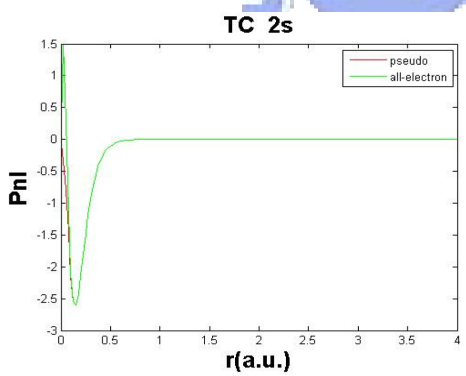


Figure 252.

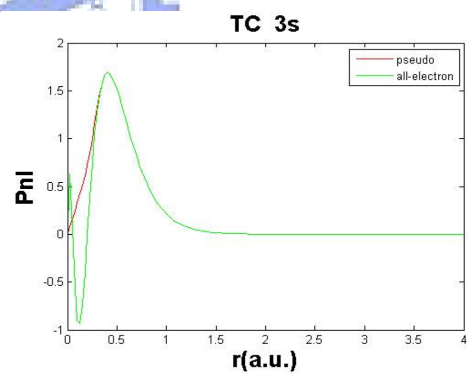


Figure 253.

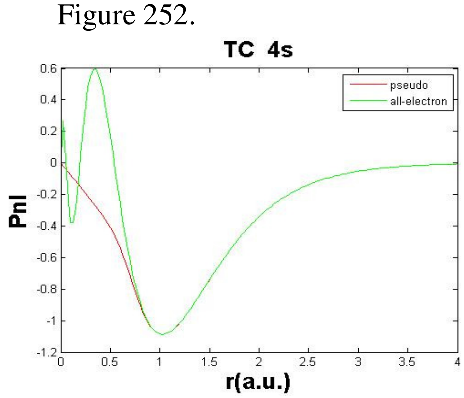


Figure 254.

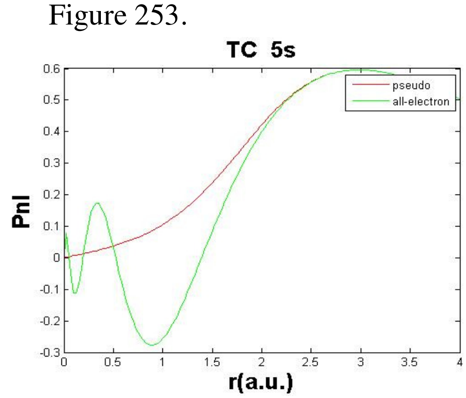


Figure 255.

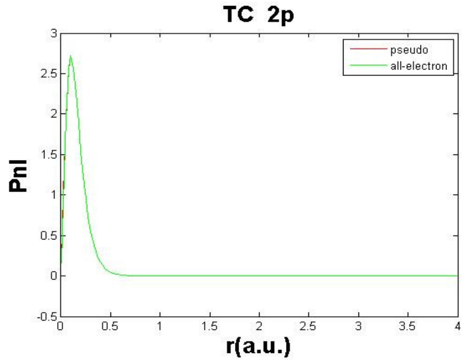


Figure 256.

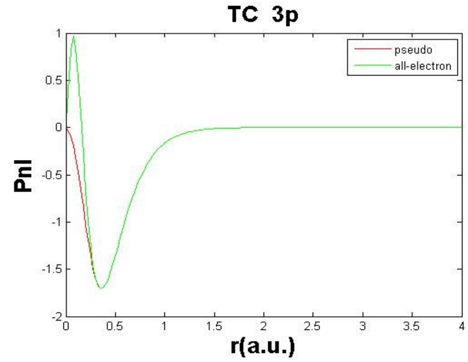


Figure 257.

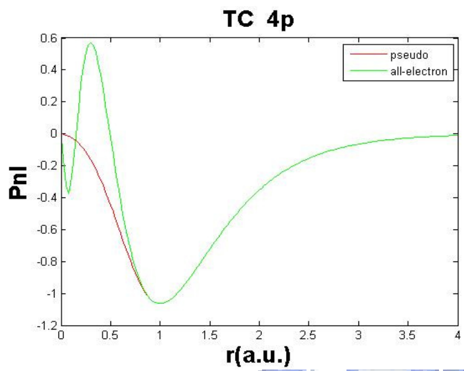


Figure 258.

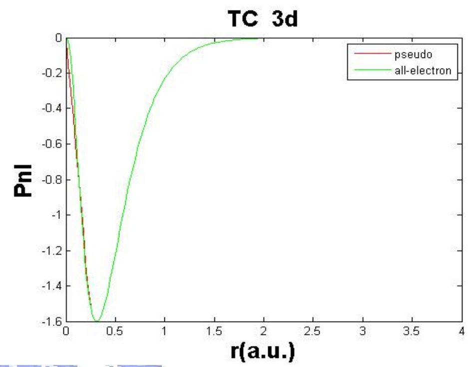


Figure 259.

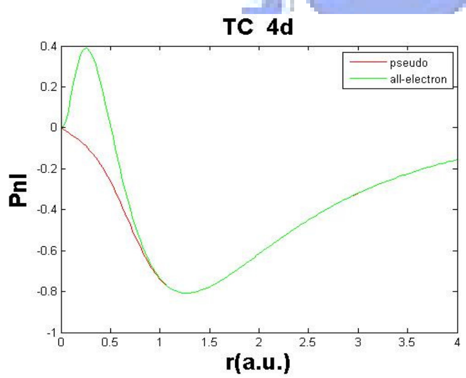


Figure 260.

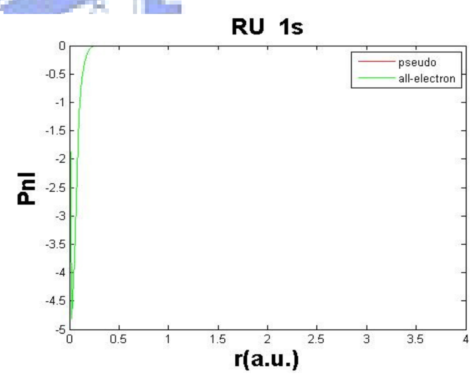


Figure 261.

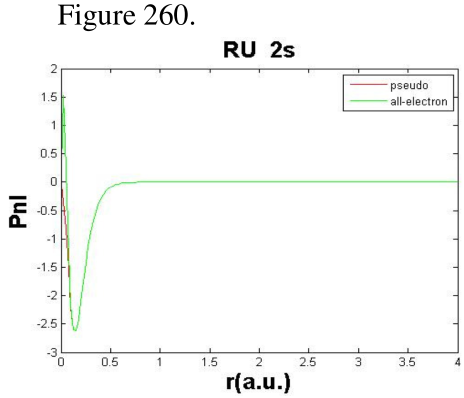


Figure 262.

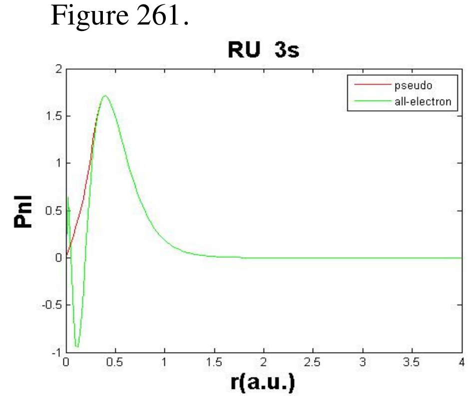


Figure 263.

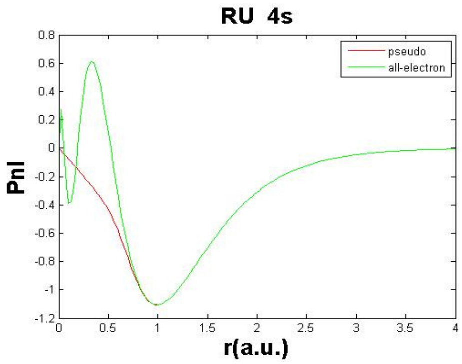


Figure 264.

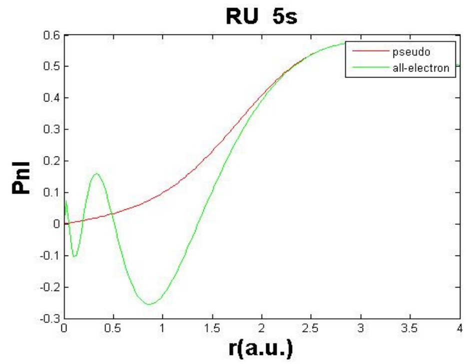


Figure 265.

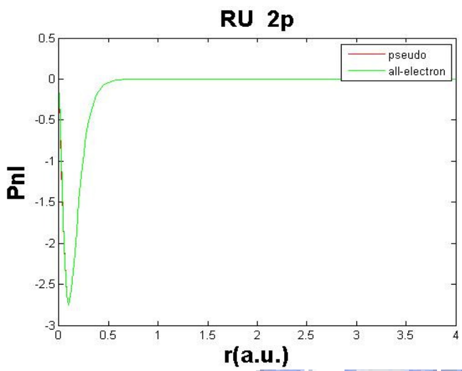


Figure 266.

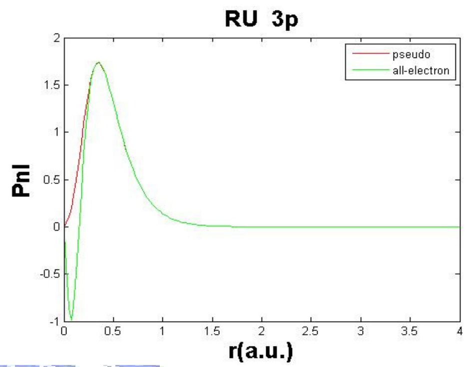


Figure 267.

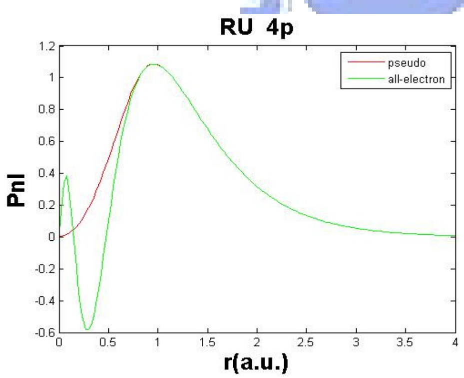


Figure 268.

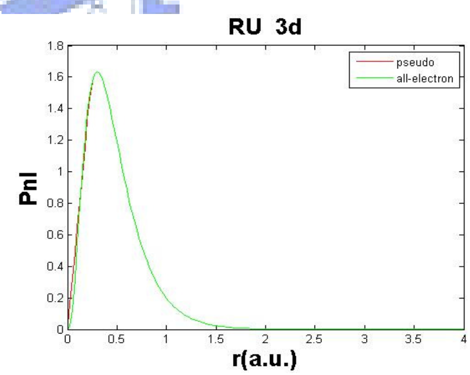


Figure 269.

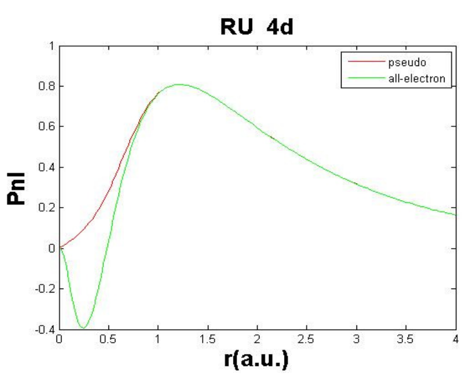


Figure 270.

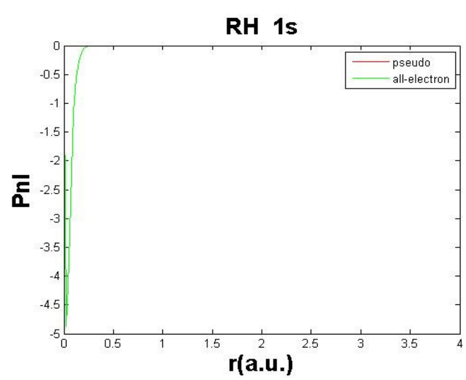


Figure 271.

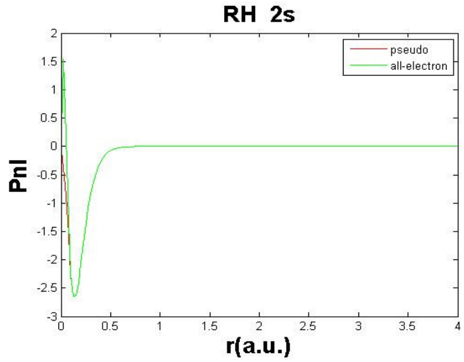


Figure 272.

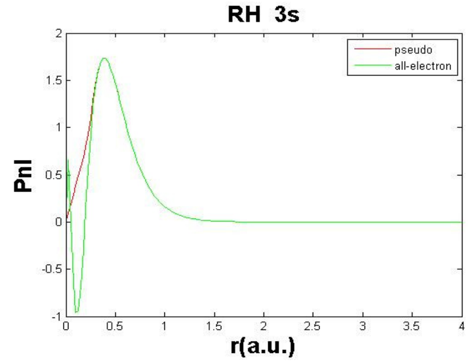


Figure 273.

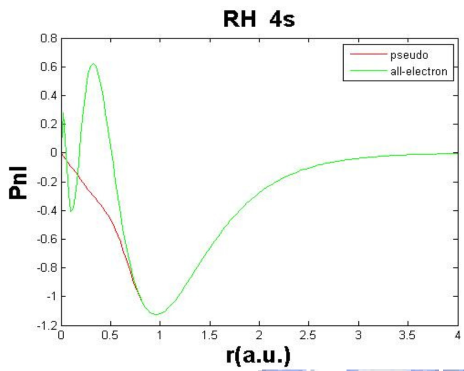


Figure 274.

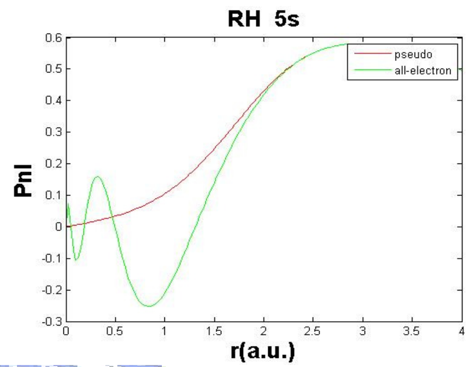


Figure 275.

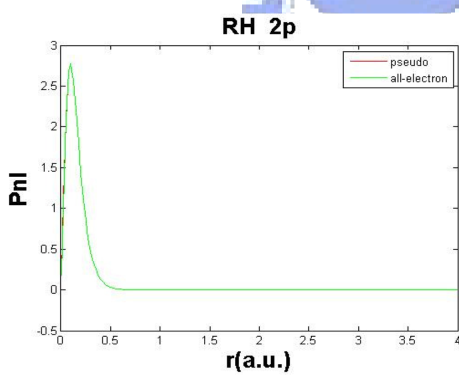


Figure 276.

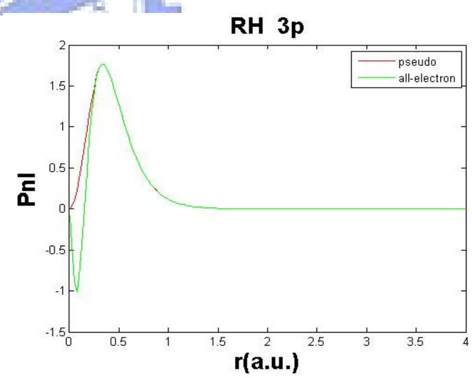


Figure 277.

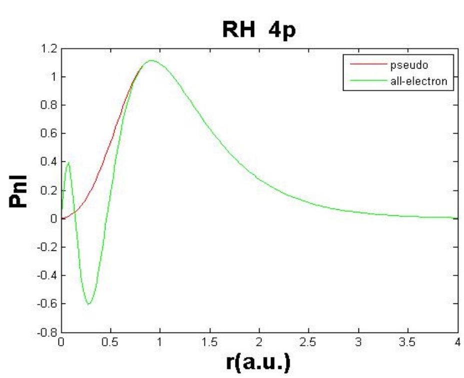


Figure 278.

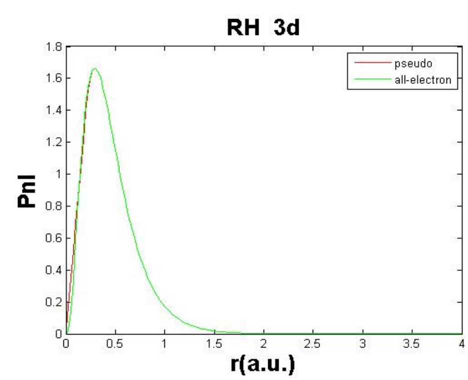


Figure 279.

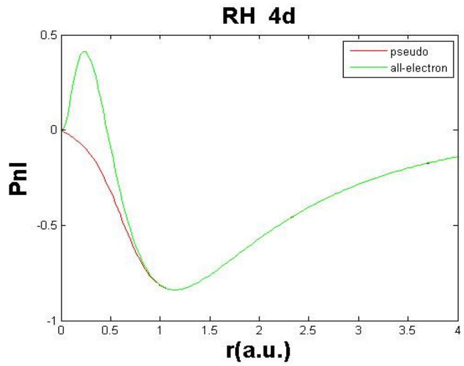


Figure 280.

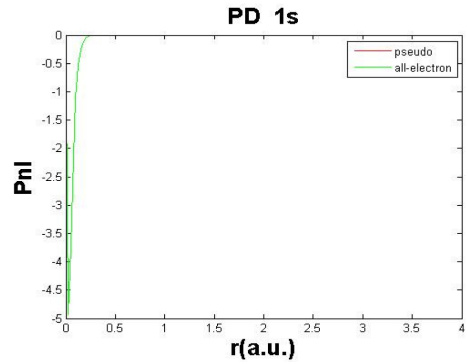


Figure 281.

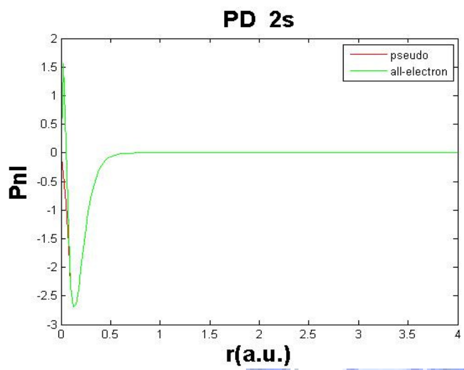


Figure 282.

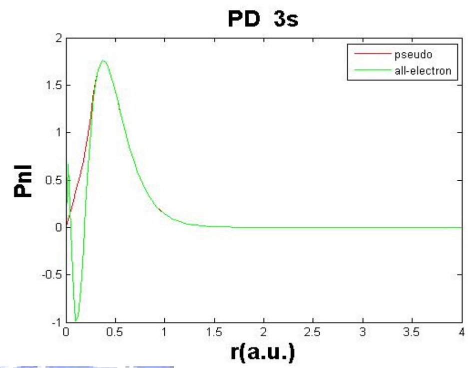


Figure 283.

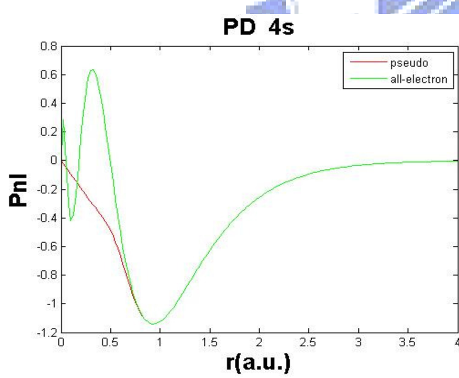


Figure 284.

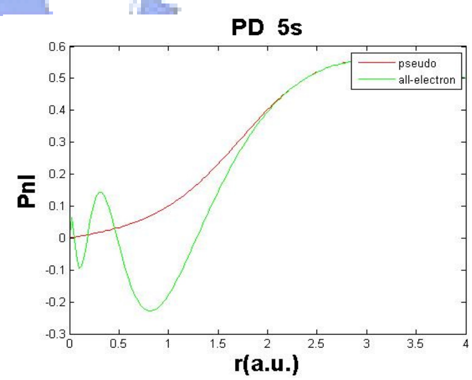


Figure 285.

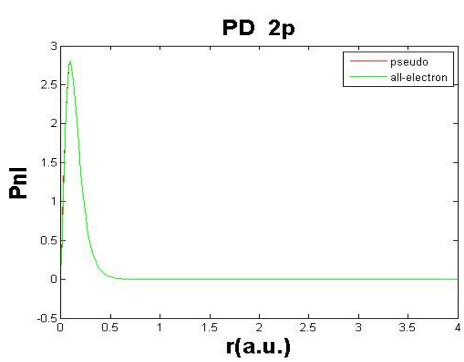


Figure 286.

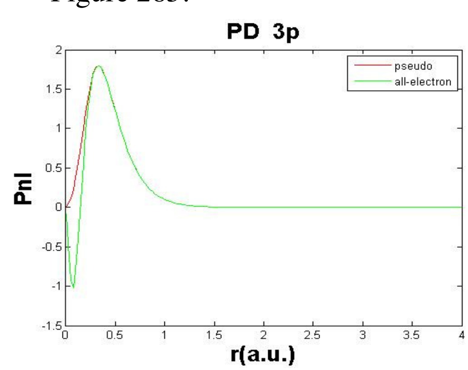


Figure 287.

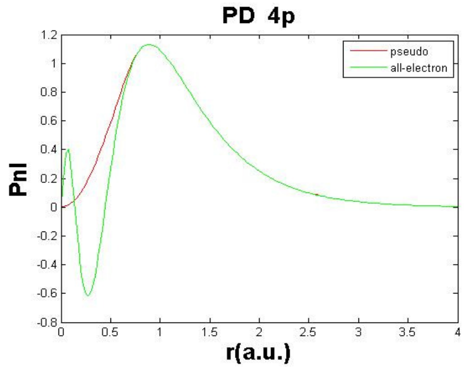


Figure 288.

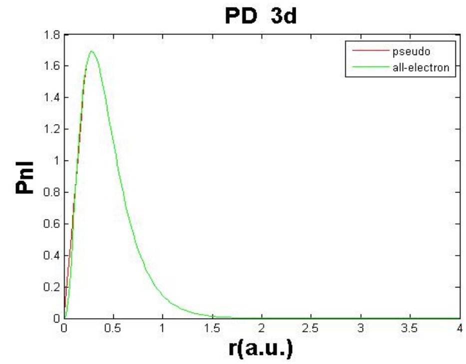


Figure 289.

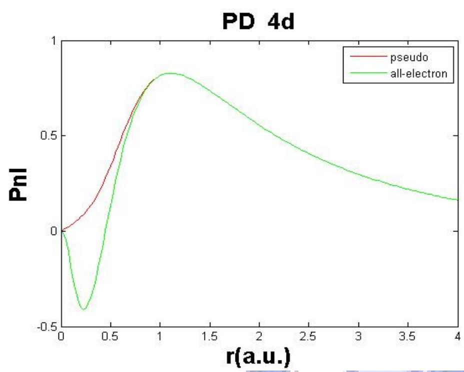


Figure 290.

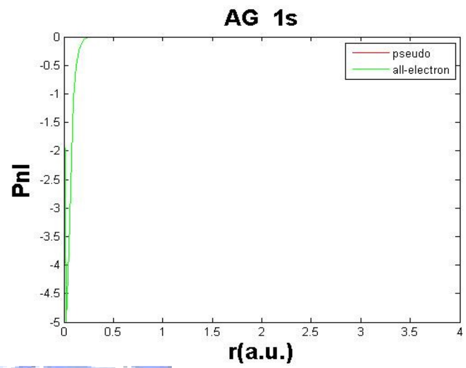


Figure 291.

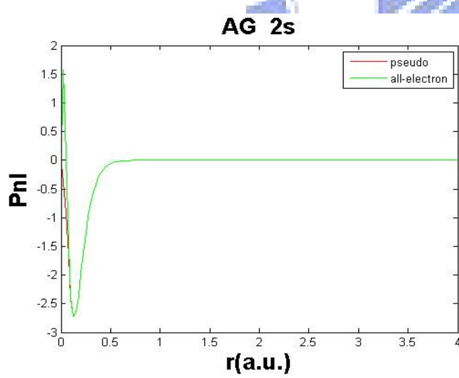


Figure 292.

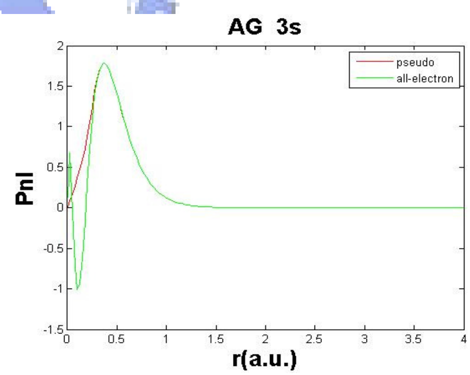


Figure 293.

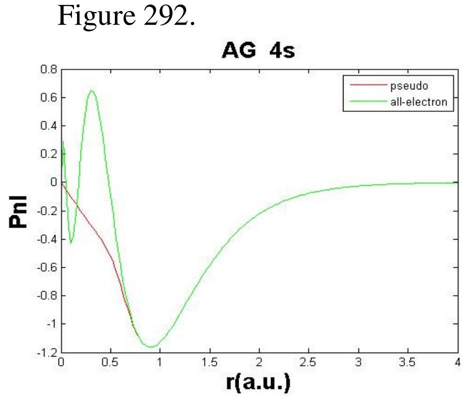


Figure 294.

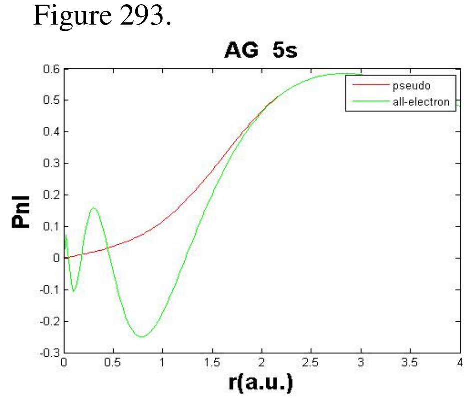


Figure 295.

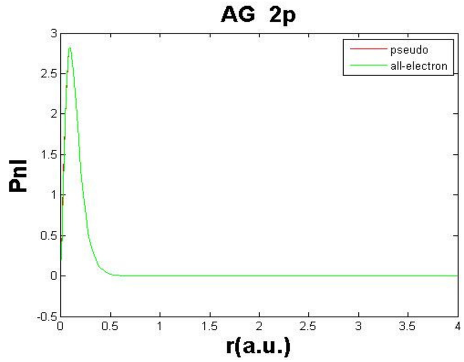


Figure 296.

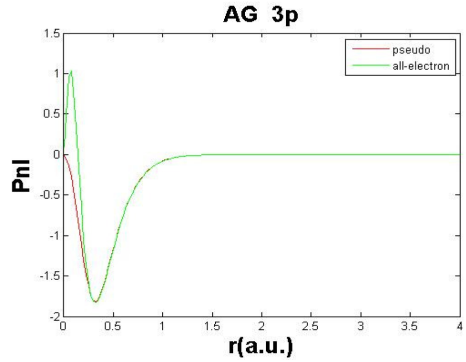


Figure 297.

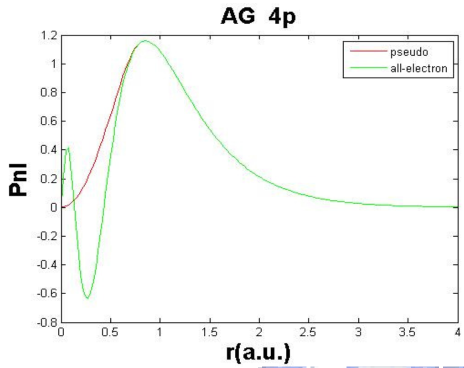


Figure 298.

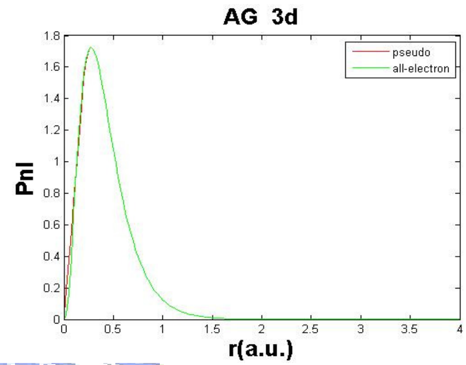


Figure 299.

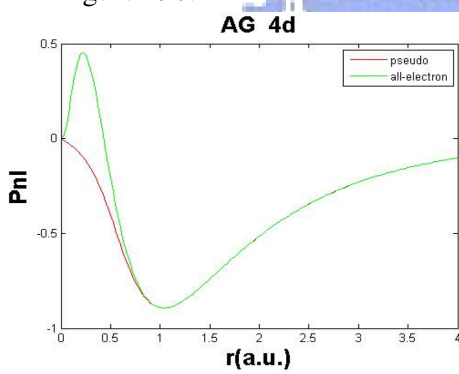
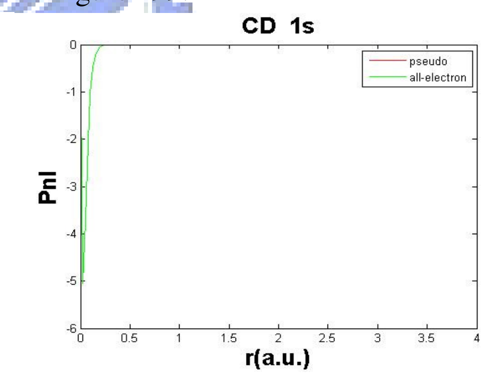


Figure 300.



Fiqure 301.

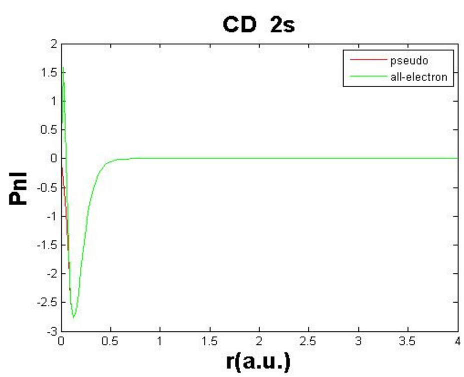


Figure 302.

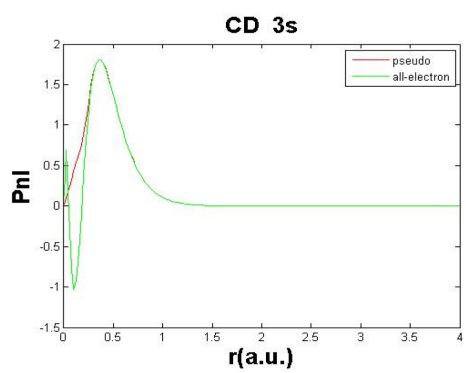


Figure 303.

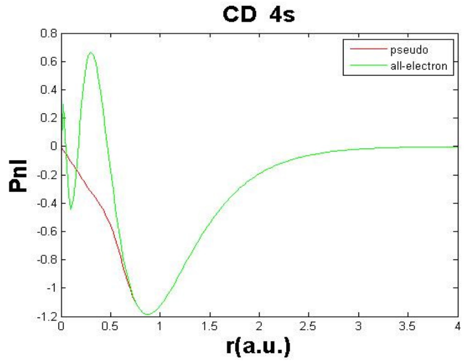


Figure 304.

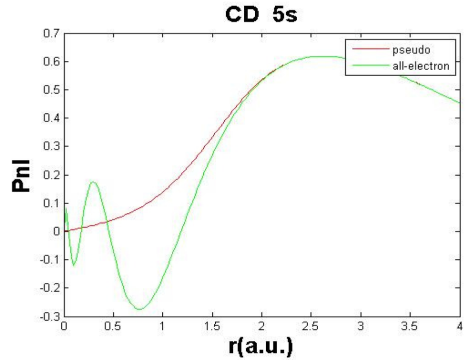


Figure 305.

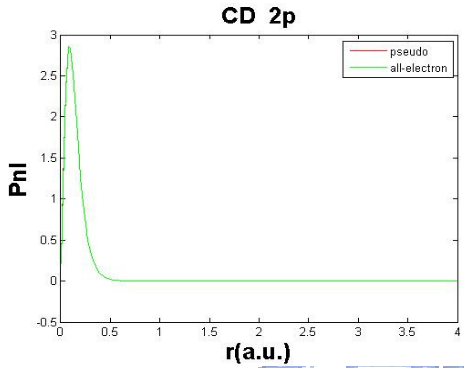


Figure 306.

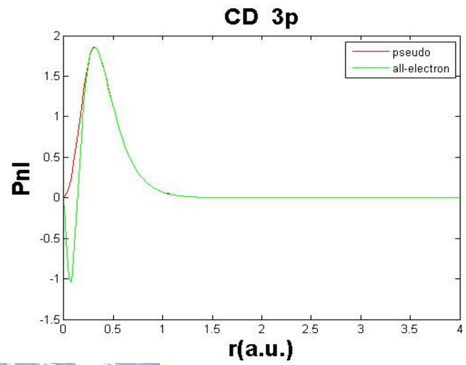


Figure 307.

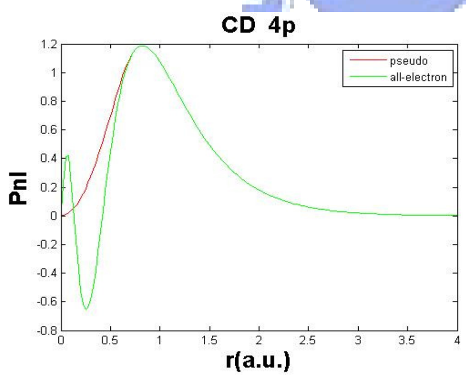


Figure 308.

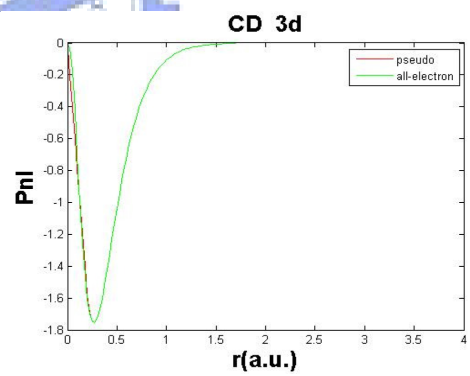


Figure 309.

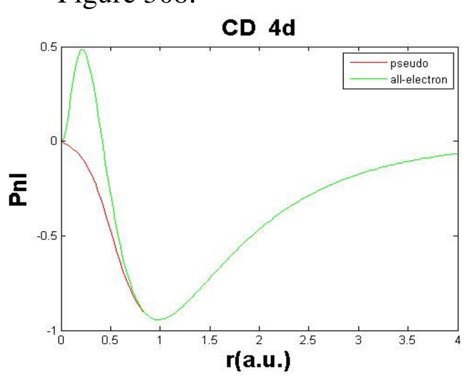


Figure 310.

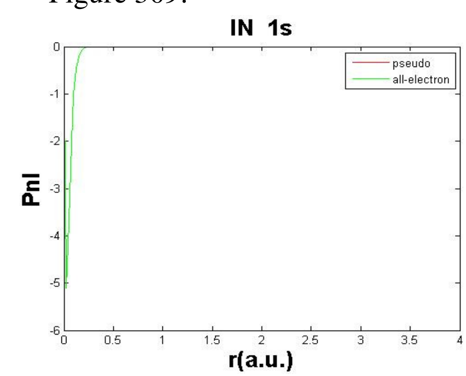


Figure 311.

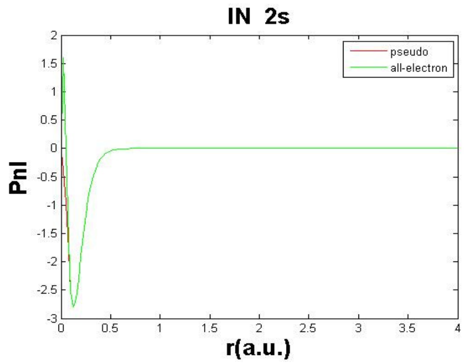


Figure 312.

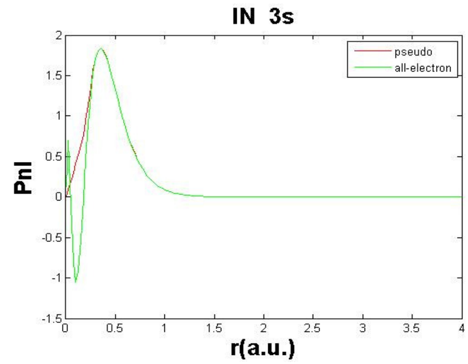


Figure313.

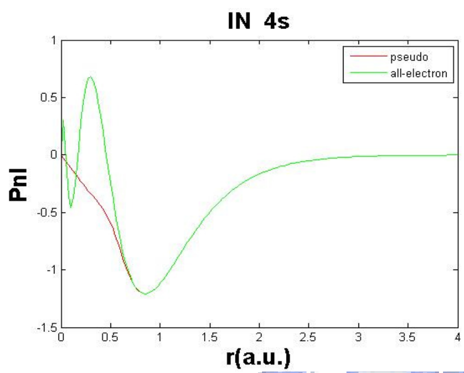


Figure 314.

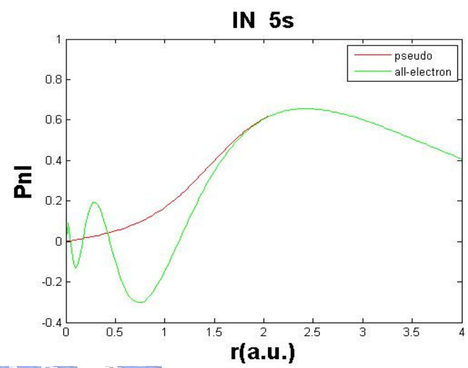


Figure 315.

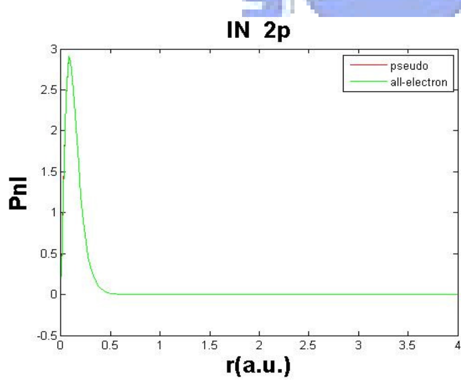


Figure 316.

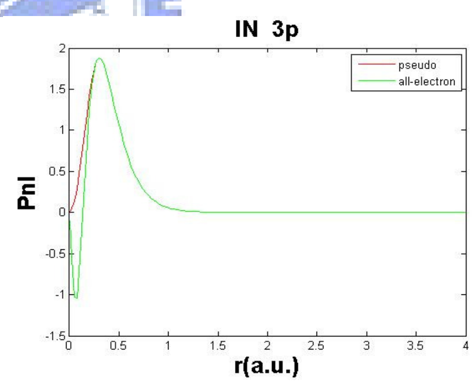


Figure 317.

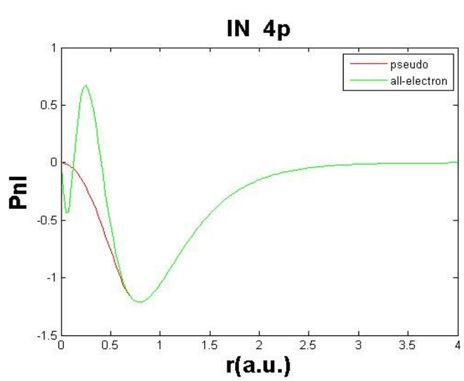


Figure 318.

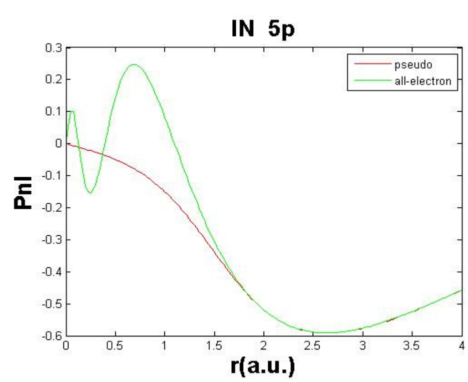


Figure 319.

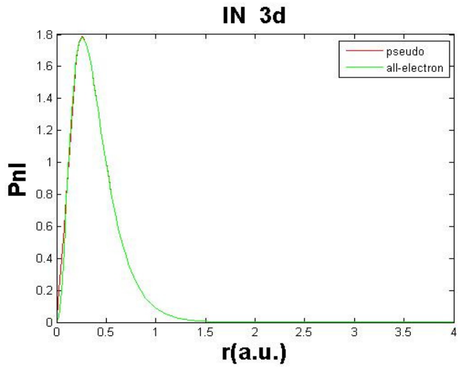


Figure 320.

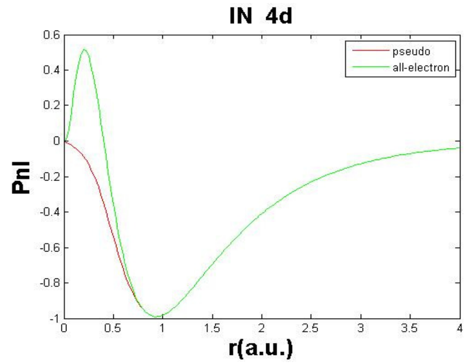


Figure 321.

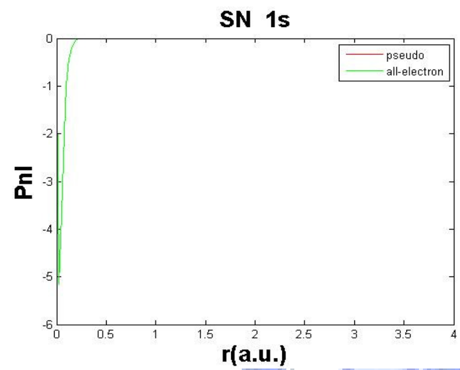


Figure 322.

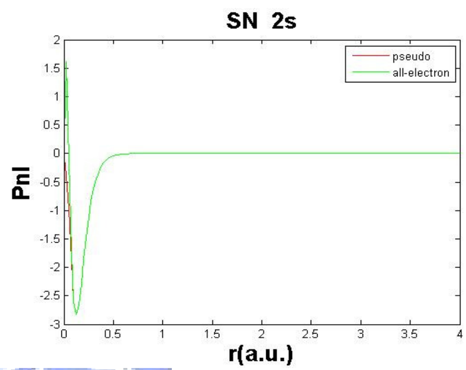


Figure 323.

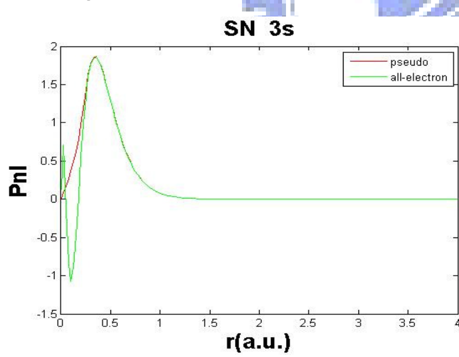


Figure 324.

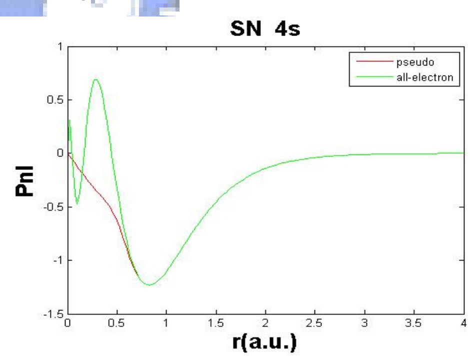


Figure 325.

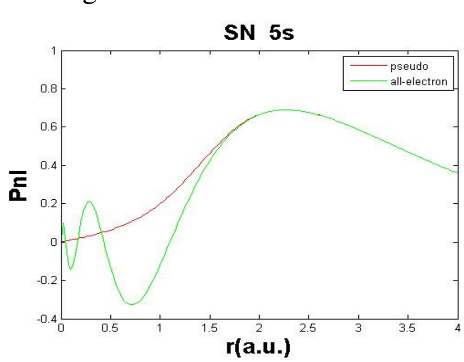


Figure 326.

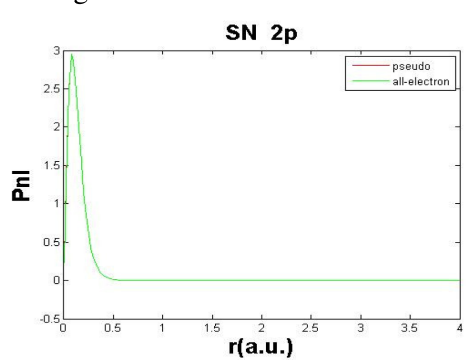


Figure 327.

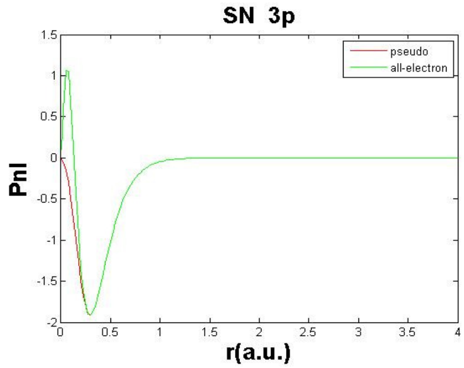


Figure 328.

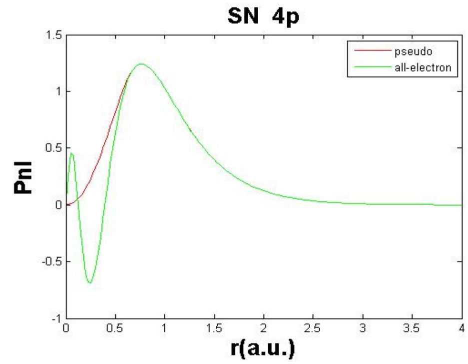


Figure 329.

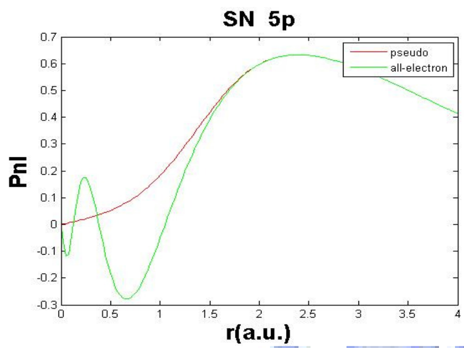


Figure 330.

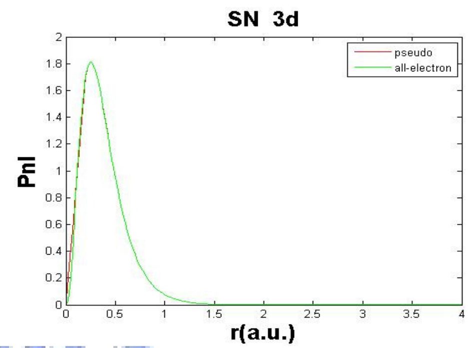


Figure 331.

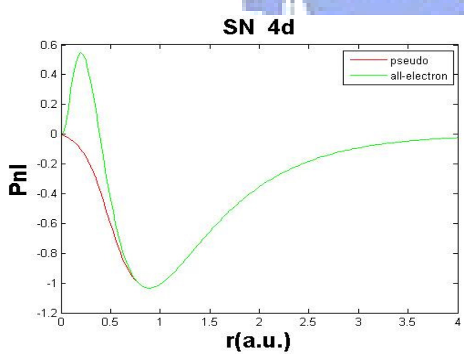


Figure 332.

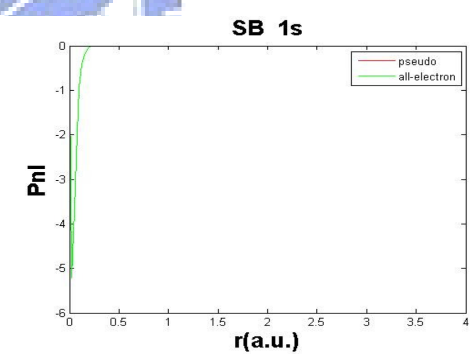


Figure 333.

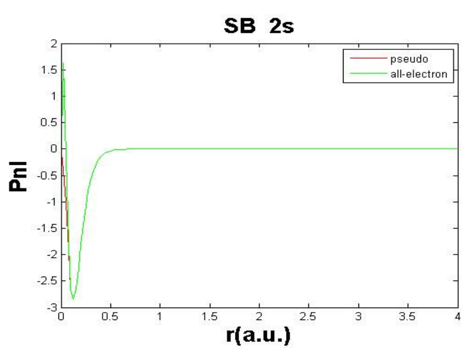


Figure 334.

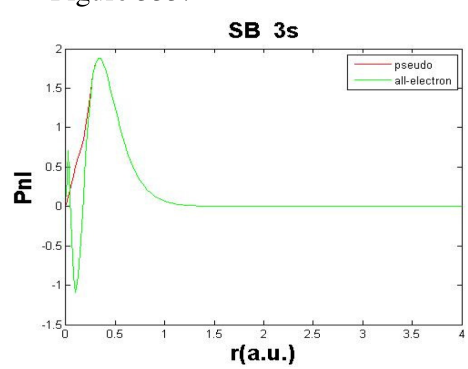


Figure 335.

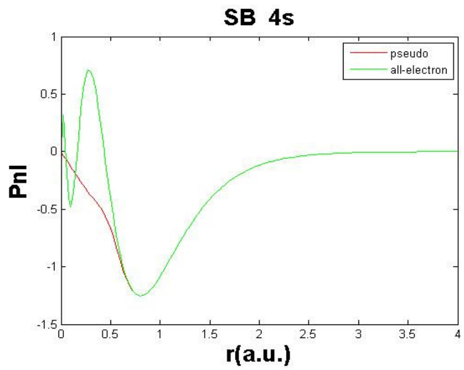


Figure 336.

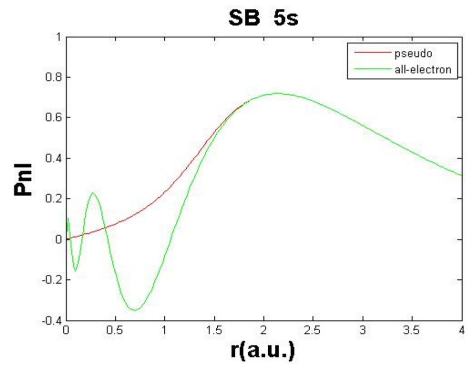


Figure 337.

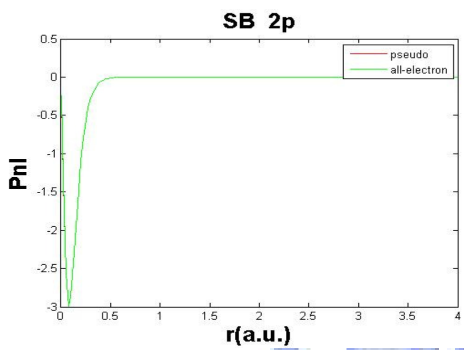


Figure 338.

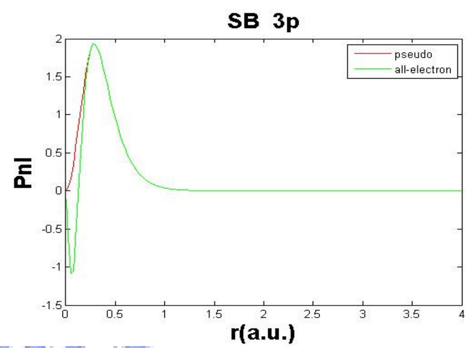


Figure 339.

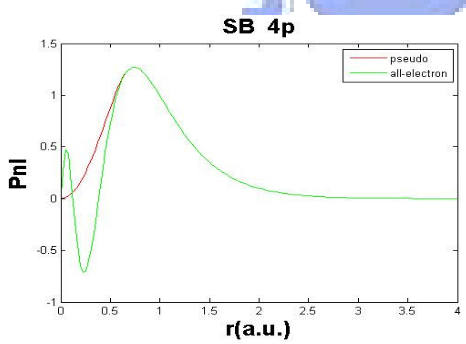


Figure 340.

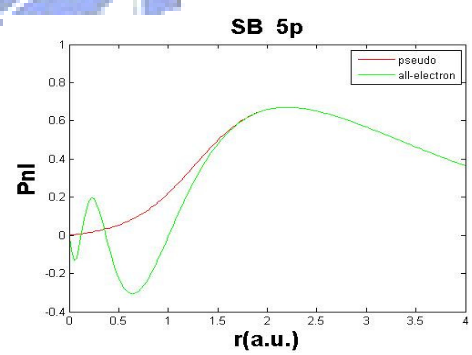


Figure 341.

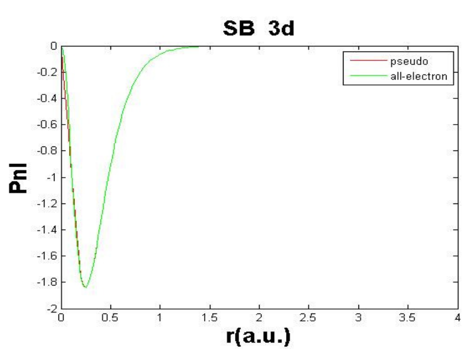


Figure 342.

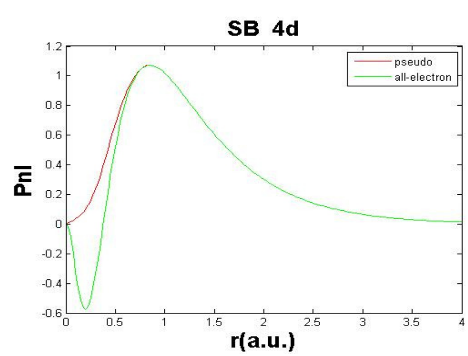


Figure 343.

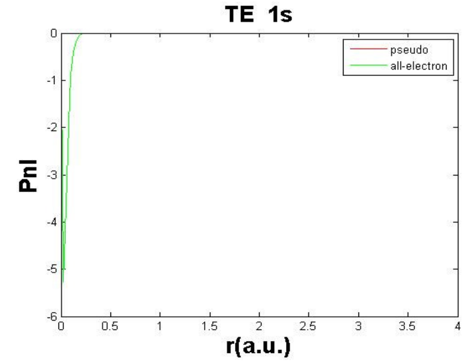


Figure 344.

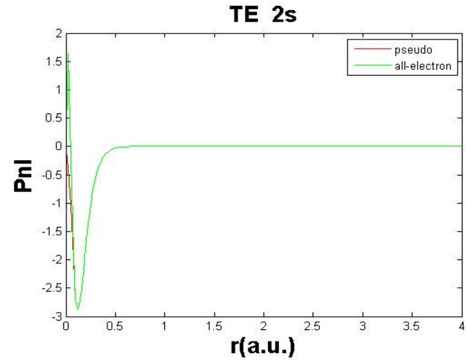


Figure 345.

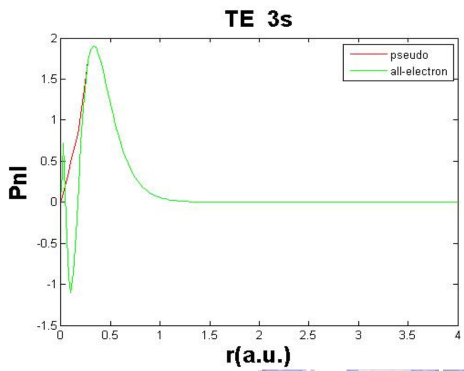


Figure   346.

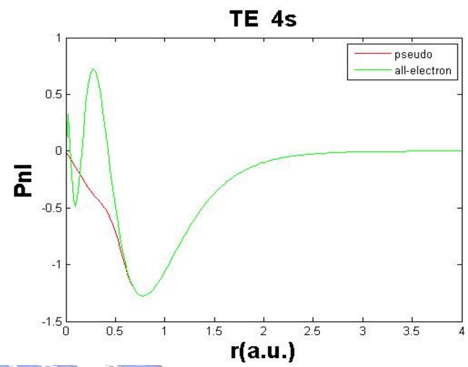


Figure 347.

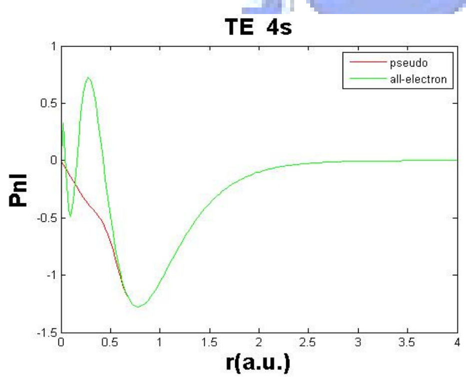


Figure 348.

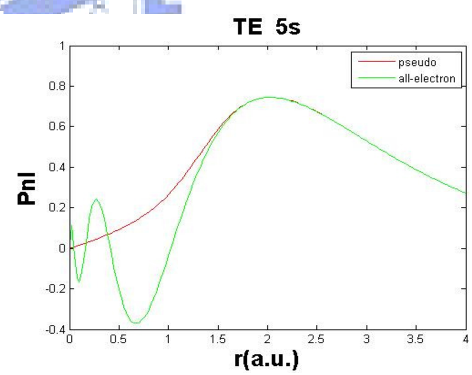


Figure 349.

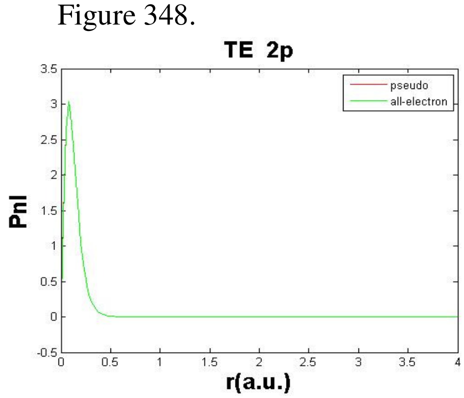


Figure 350.

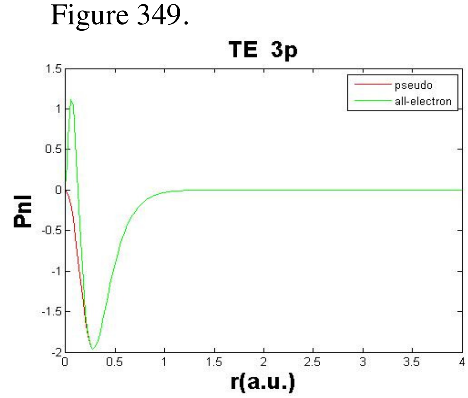


Figure 351.

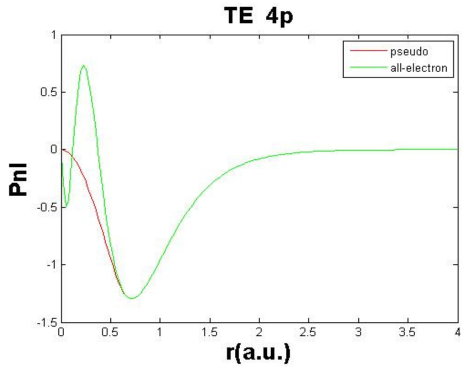


Figure 352.

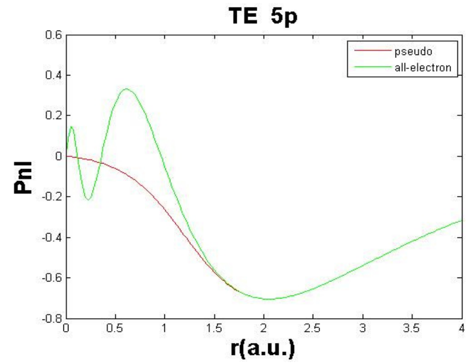


Figure 353.

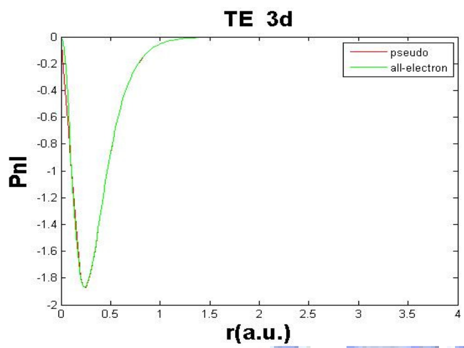


Figure 354.

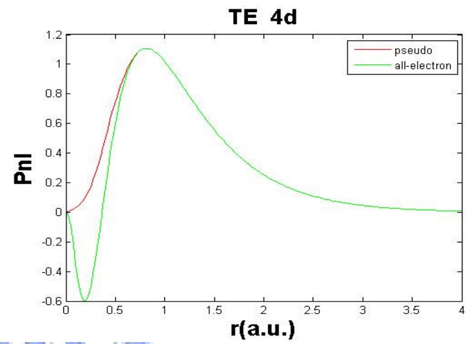


Figure 355.

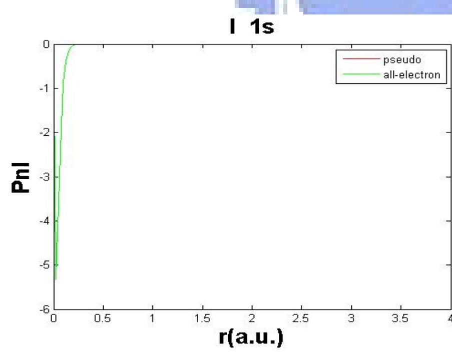


Figure 356.

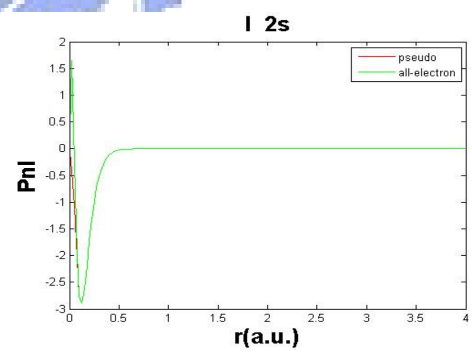


Figure 357.

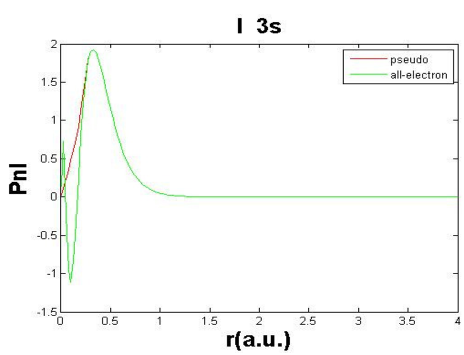


Figure 358.

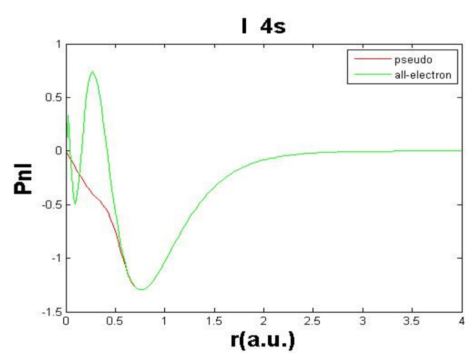


Figure 359.

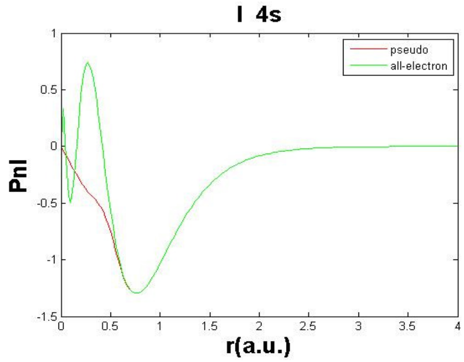


Figure 360.

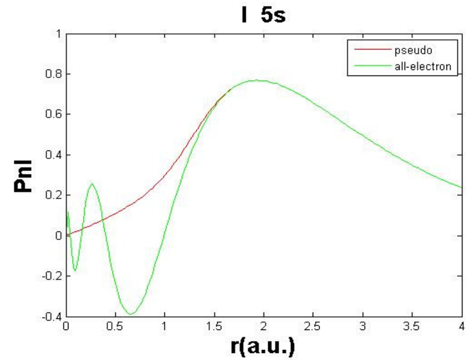


Figure 361.

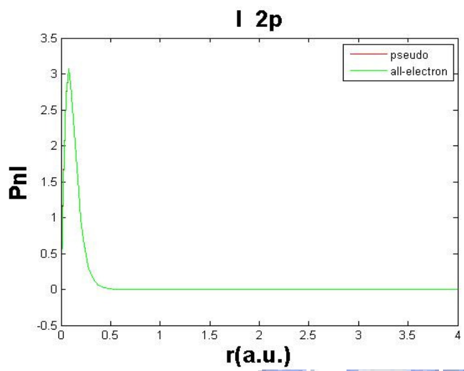


Figure 362.

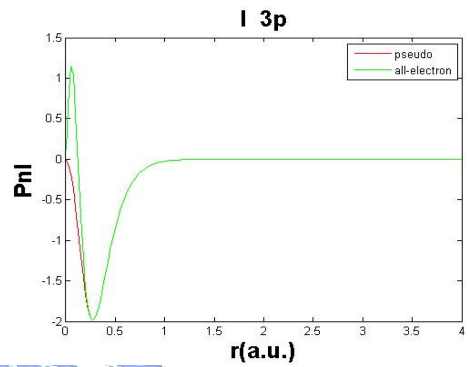


Figure 363.

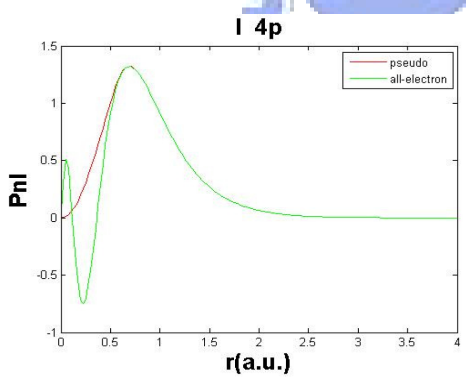


Figure 364.

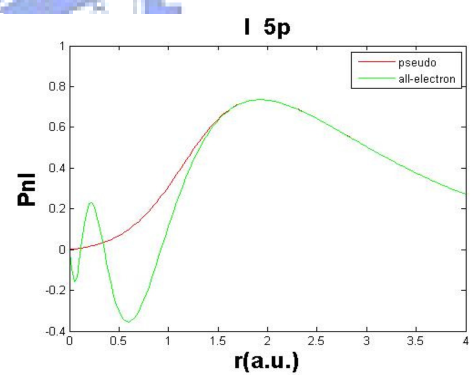


Figure 365.

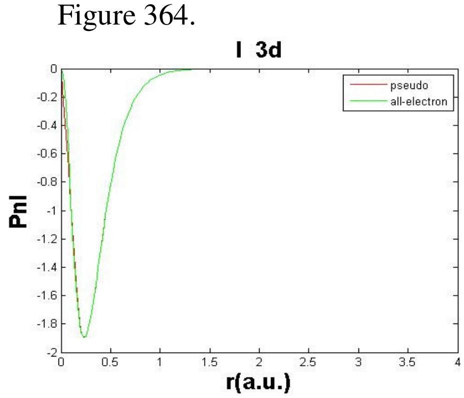


Figure 366.

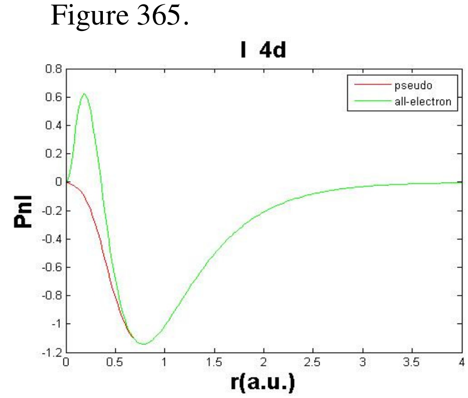


Figure 367.

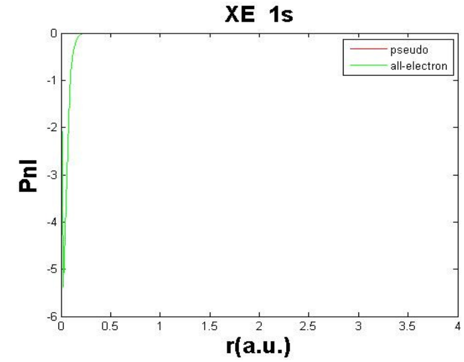


Figure 368.

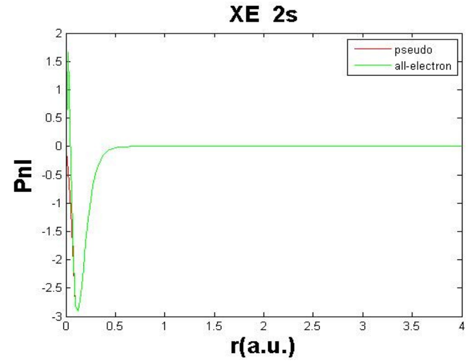


Figure 369.

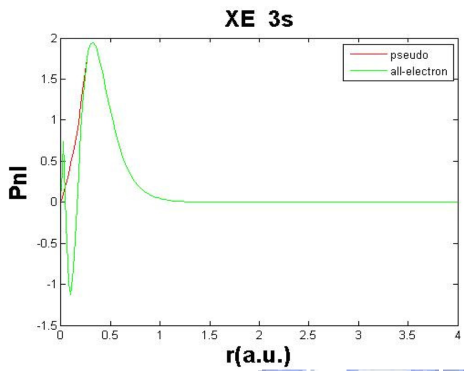


Figure 370.

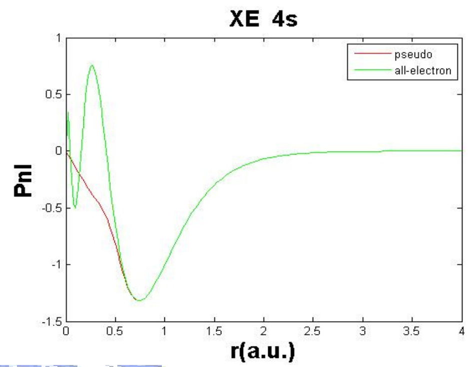


Figure 371.

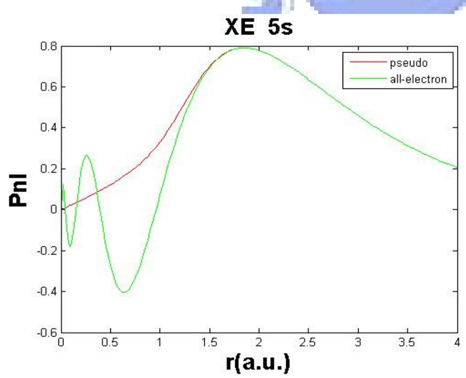


Figure 372.

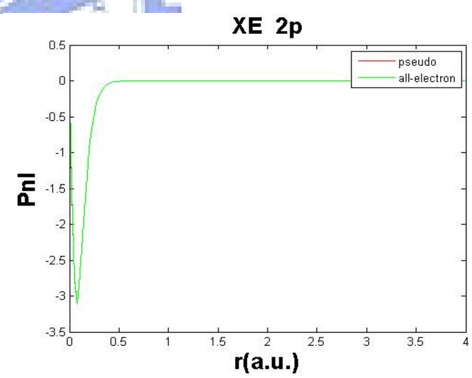


Figure 373.

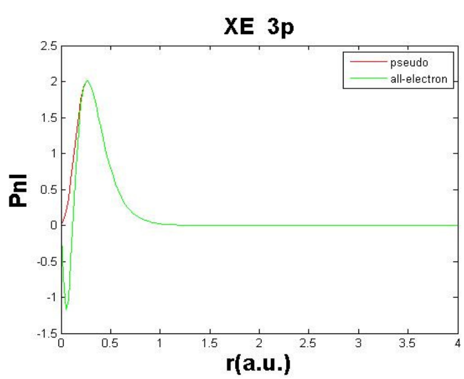


Figure 374.

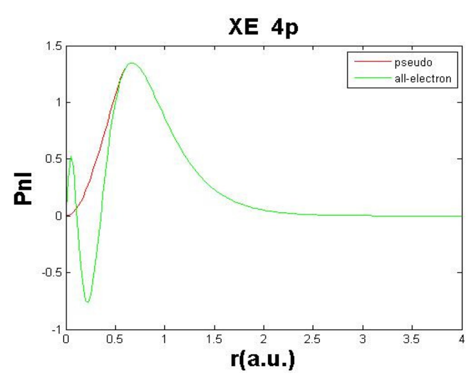


Figure 375.

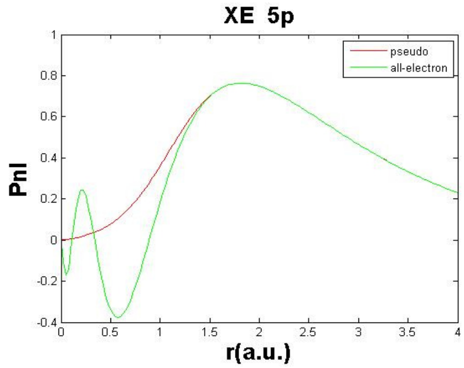


Figure 376.

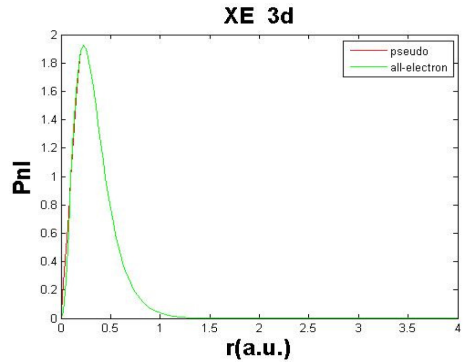


Figure 377.

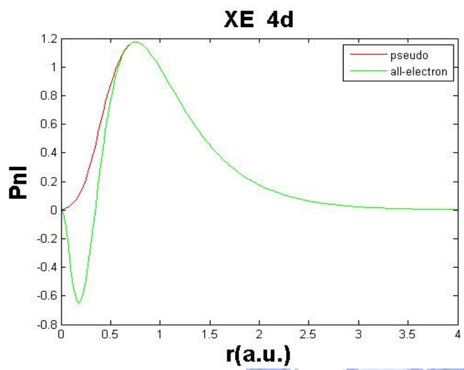


Figure 378.

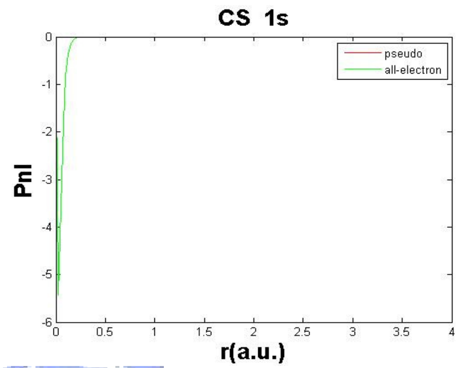


Figure 379.

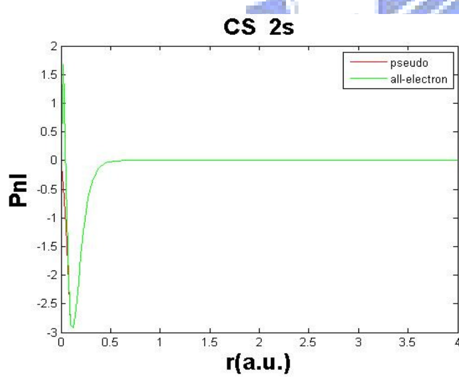


Figure 380.

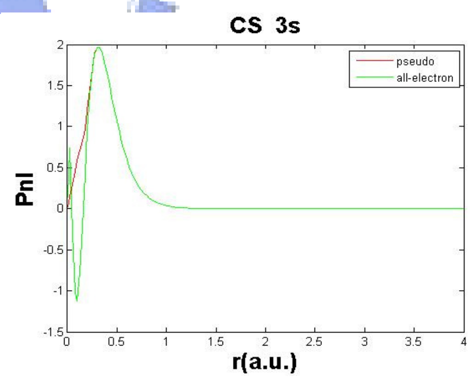


Figure 381.

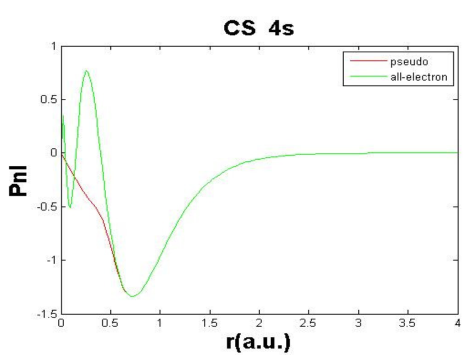


Figure 382.

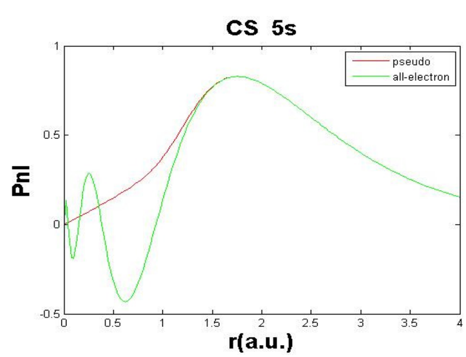


Figure 383.

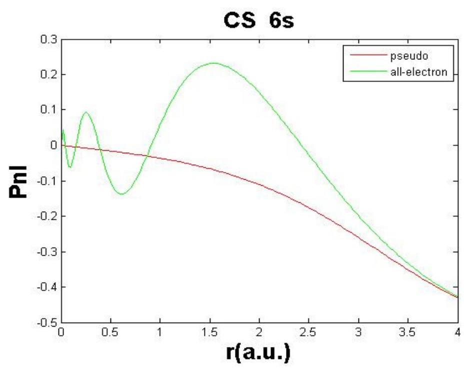


Figure 384.

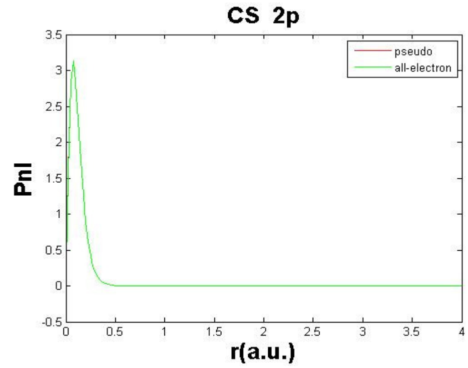


Figure 385.

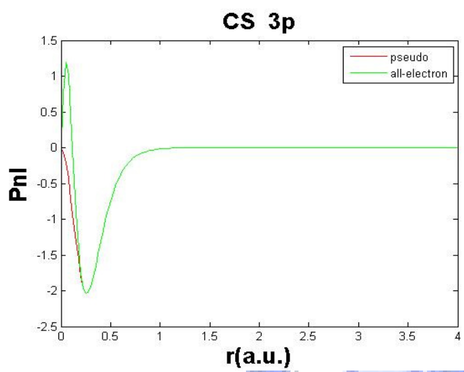


Figure 386.

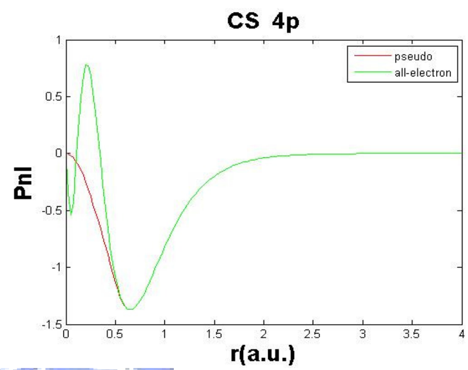


Figure 387.

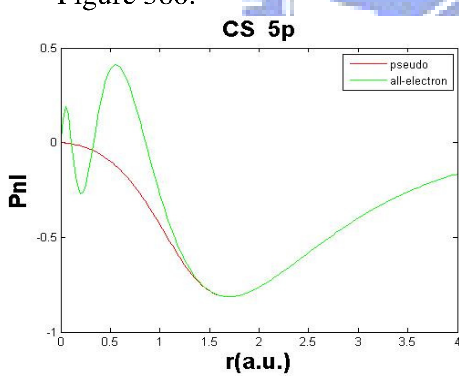


Figure 388.

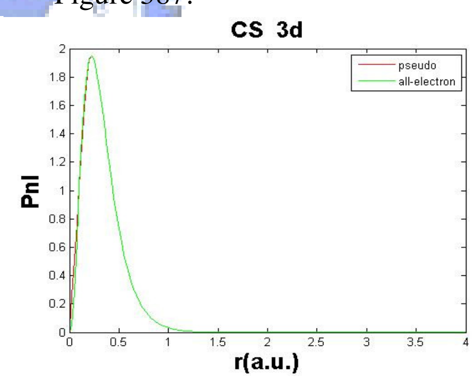


Figure 389.

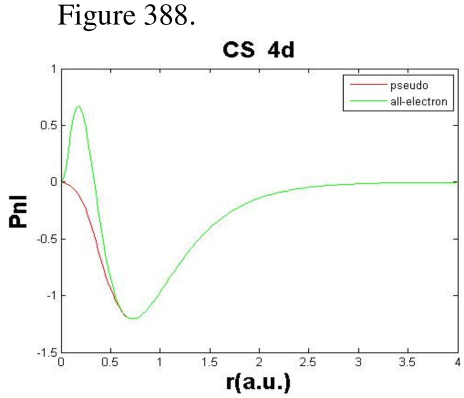


Figure 390.

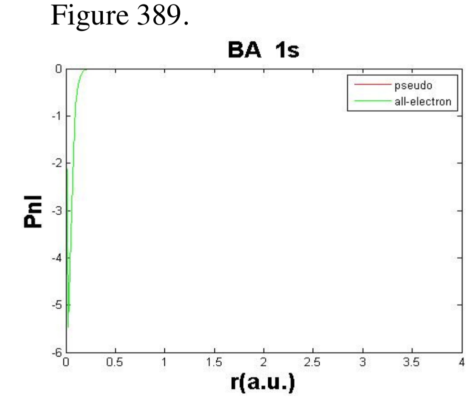


Figure 391.

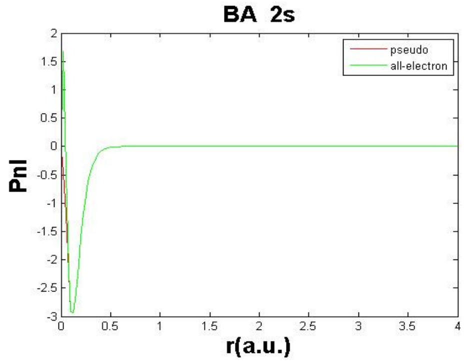


Figure 392.

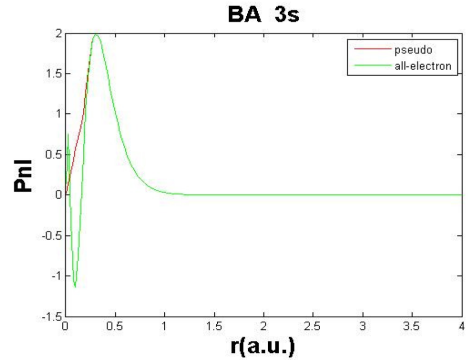


Figure 393.

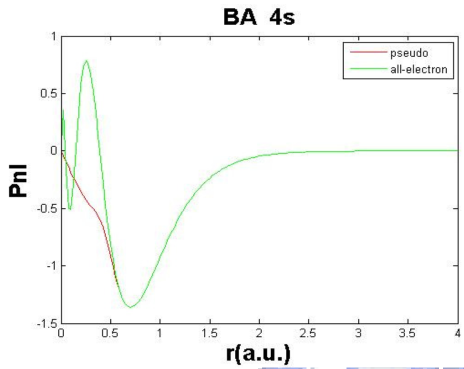


Figure 394.

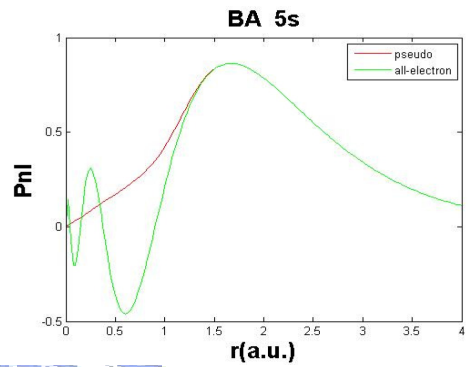


Figure 395.

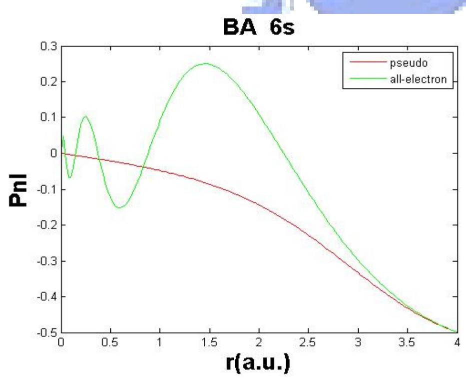


Figure 396.

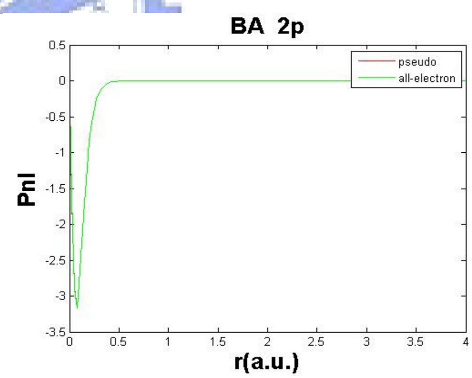


Figure 397.

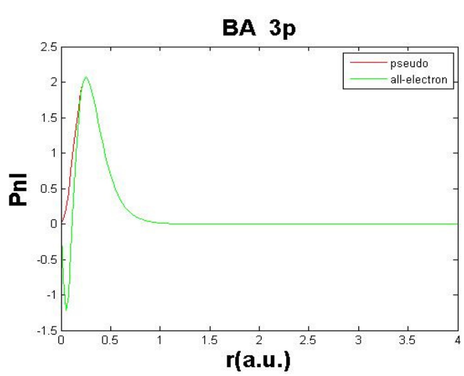


Figure 398.

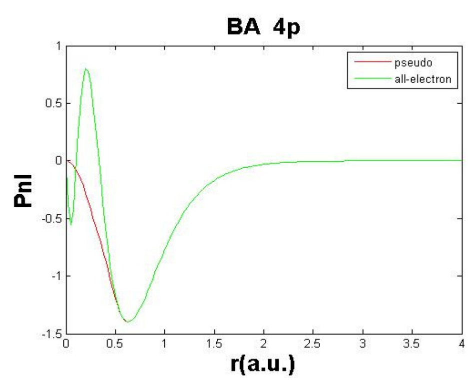


Figure 399.

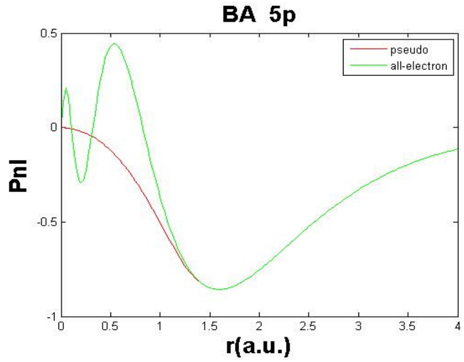


Figure 400.

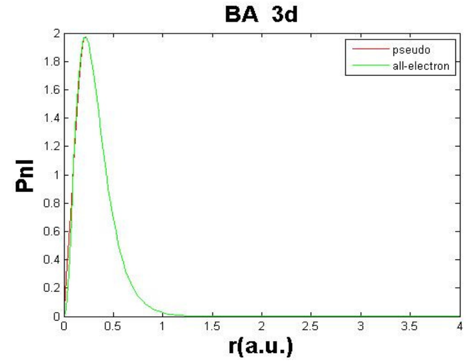


Figure 401.

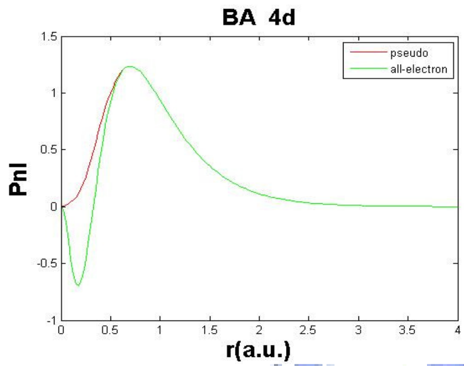


Figure 402.

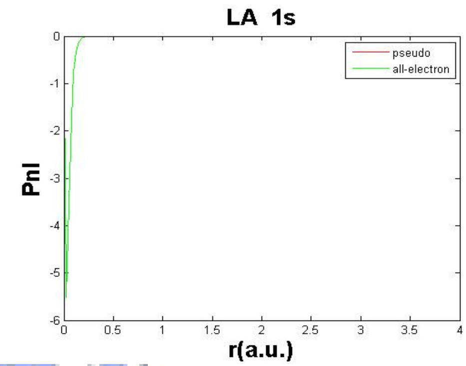


Figure 403.

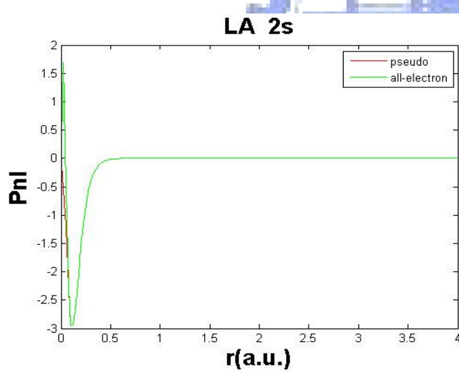


Figure 404.

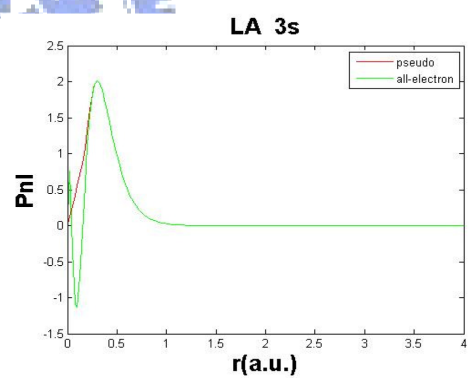


Figure 405.

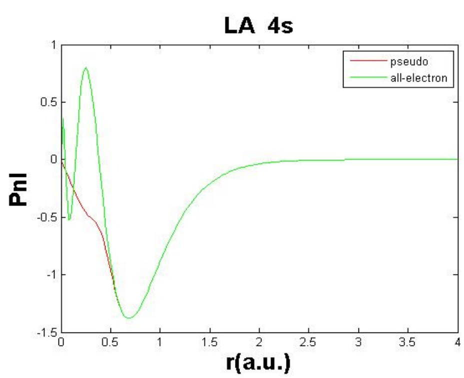


Figure 406.

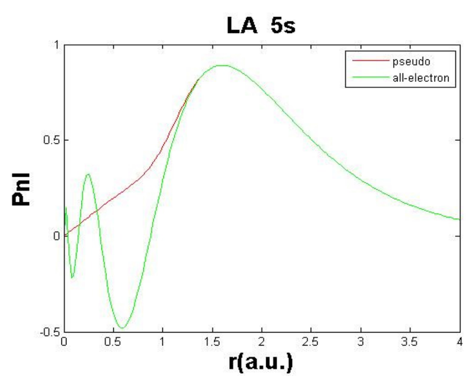


Figure 407.

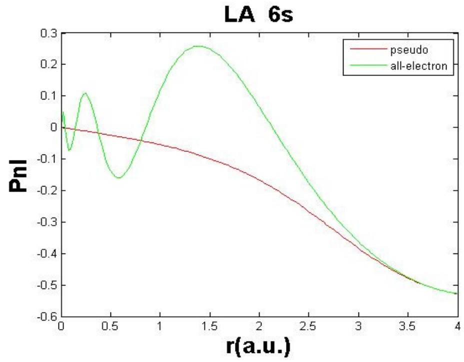


Figure 408.

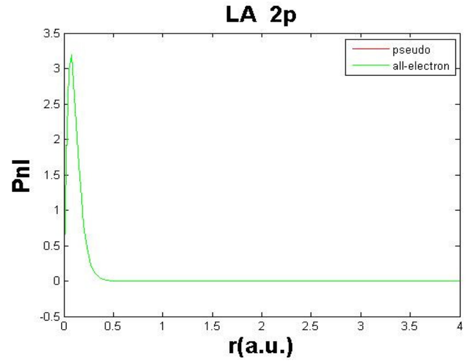


Figure 409.

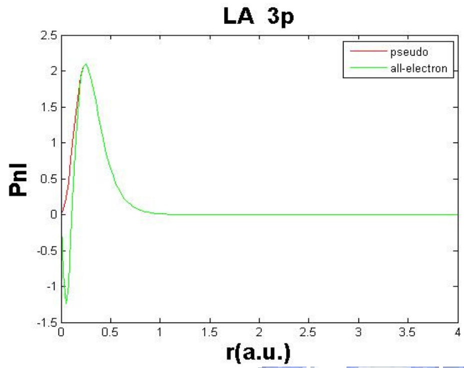


Figure 410.

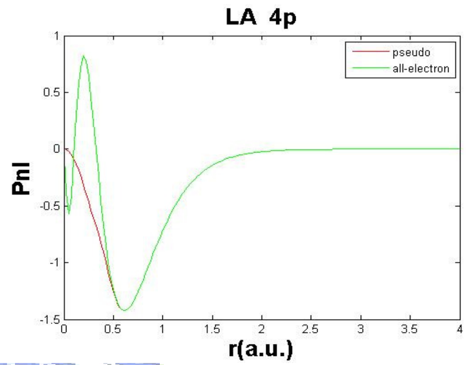


Figure 411.

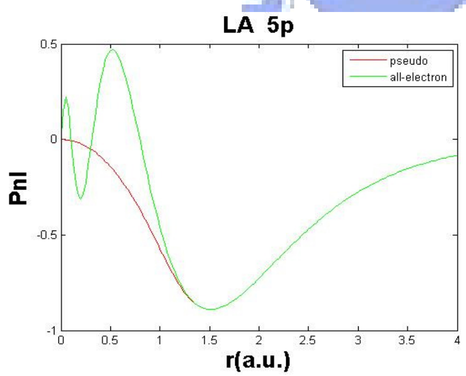


Figure 412.

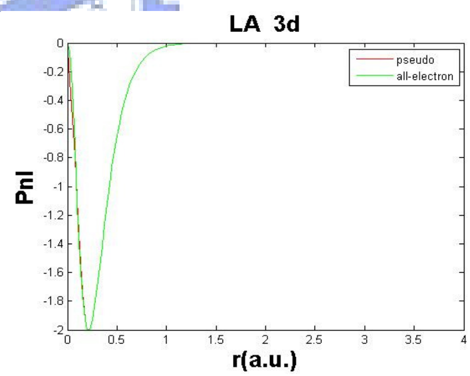


Figure 413.

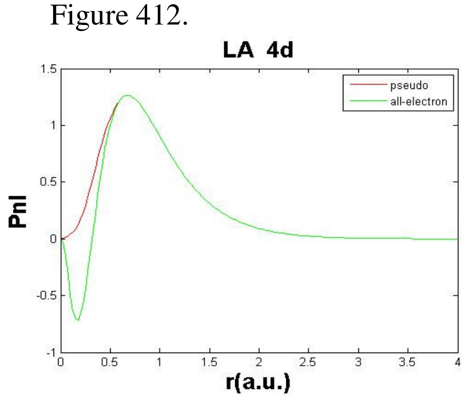


Figure 414.

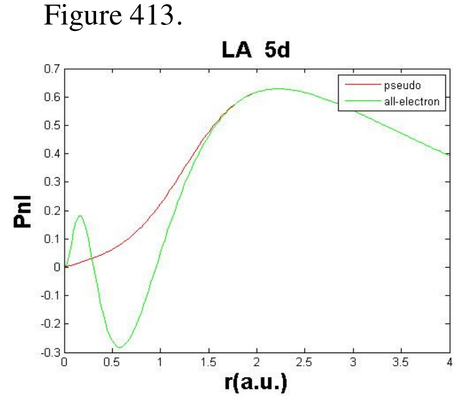


Figure 415.

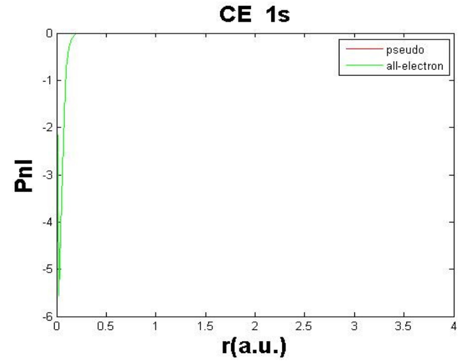


Figure 416.

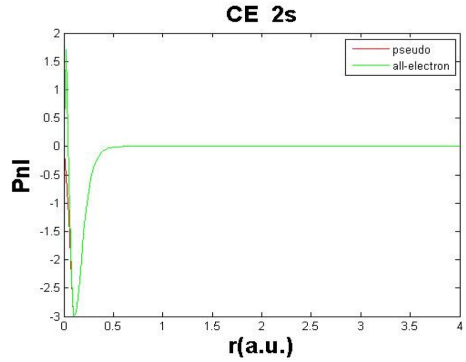


Figure 417.

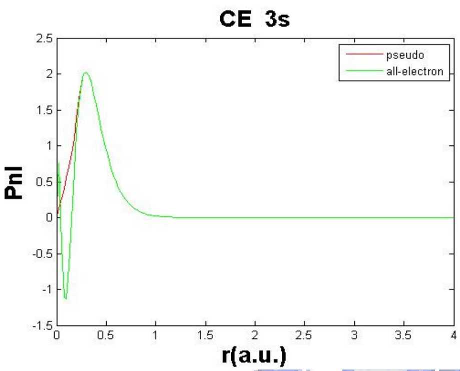


Figure 418.

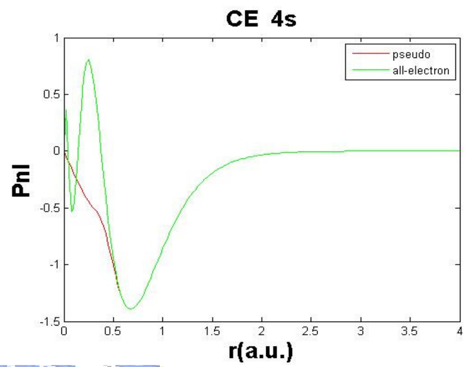


Figure 419.

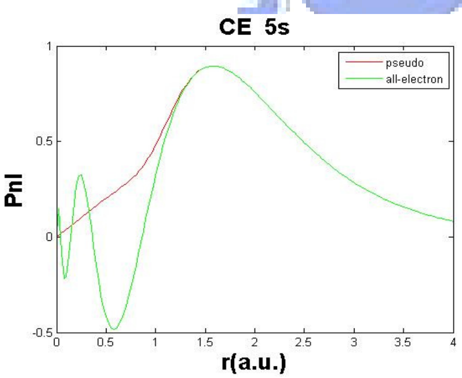


Figure 420.

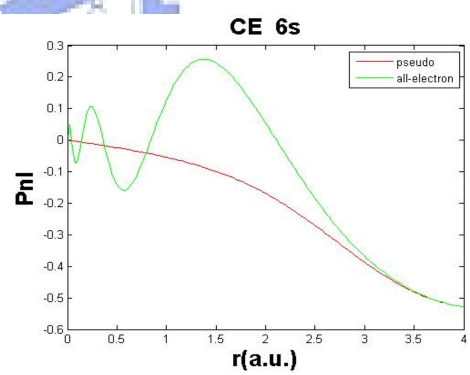


Figure 421.

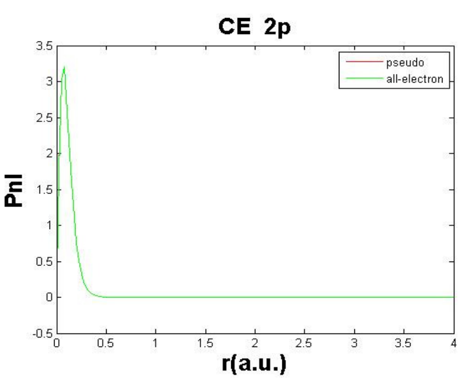


Figure 422.

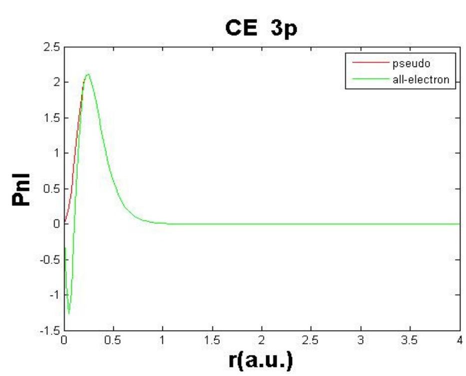


Figure 423.

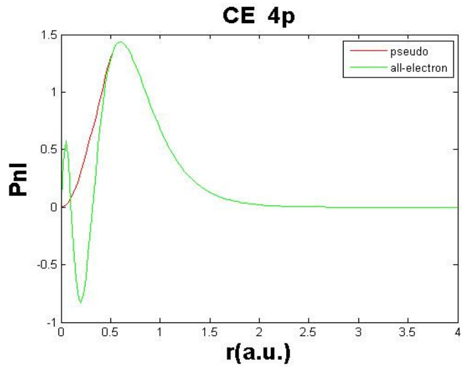


Figure 424.

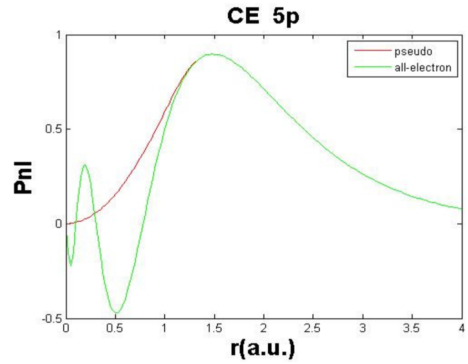


Figure 425.

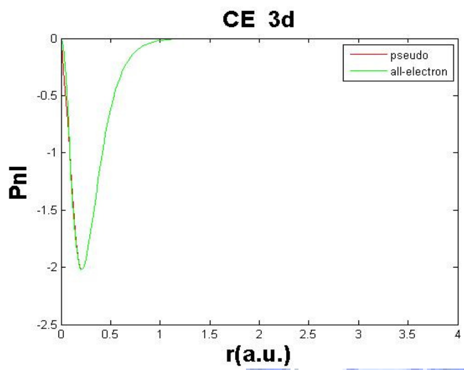


Figure 426.

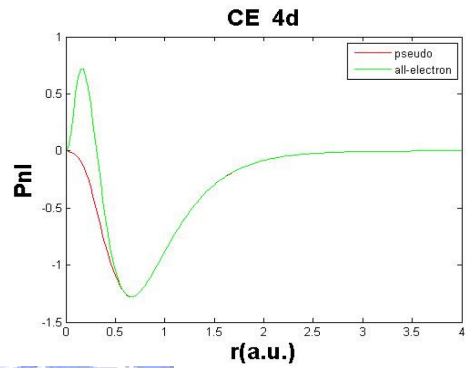


Figure 427.

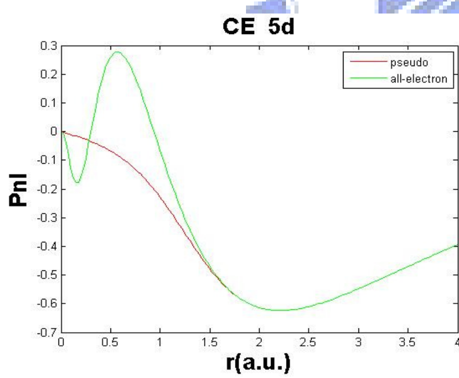


Figure 428.

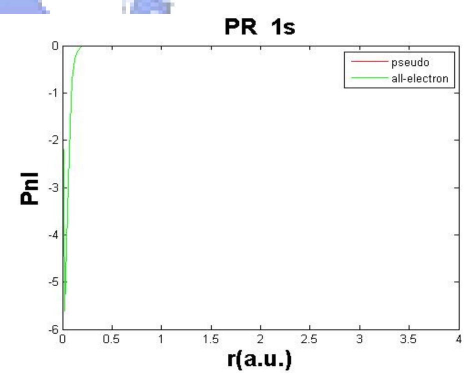


Figure 429.

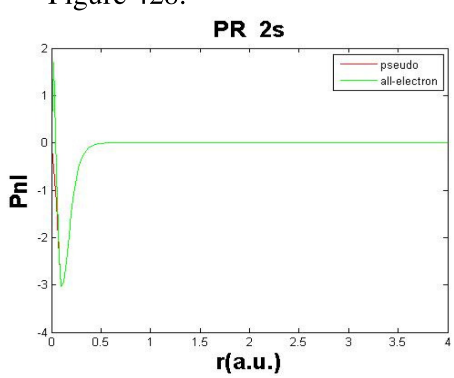


Figure 430.

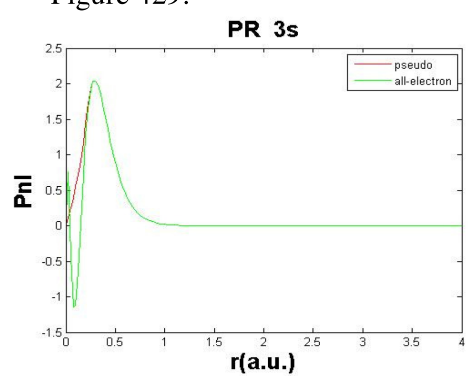


Figure 431.

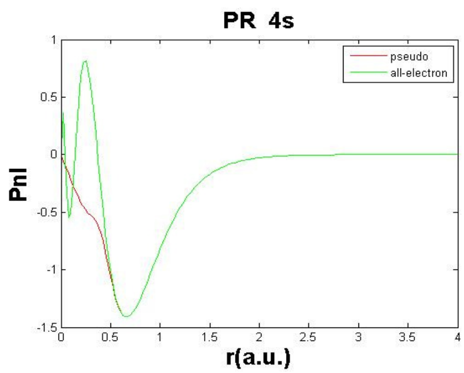


Figure 432.

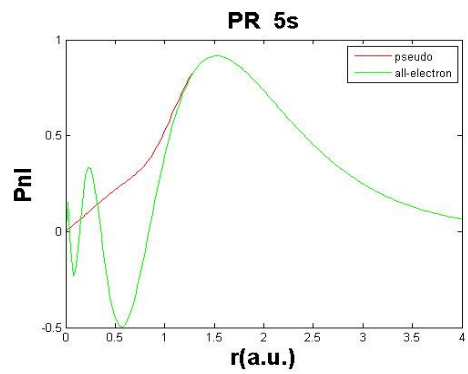


Figure 433.

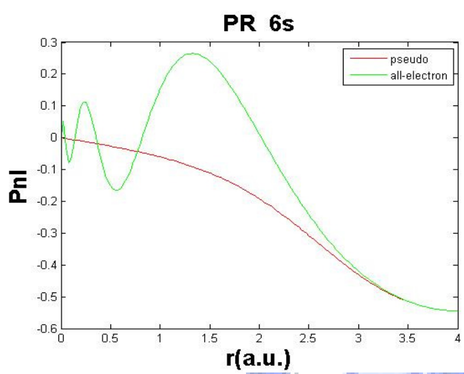


Figure 434.

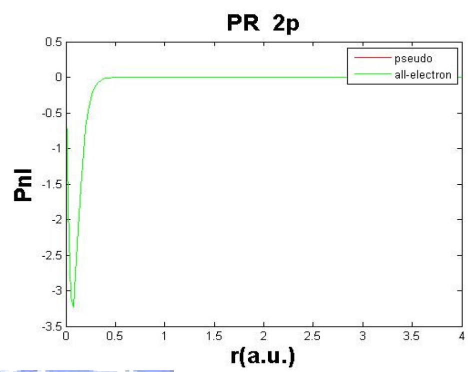


Figure 435.

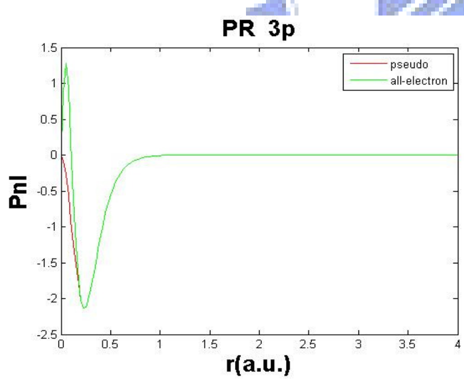


Figure 436.

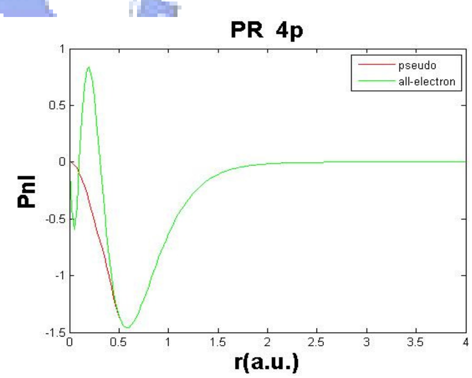


Figure 437.

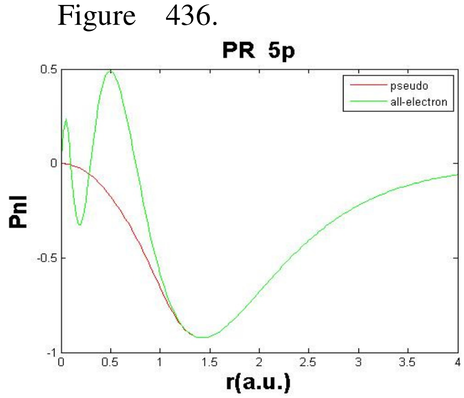


Figure 438.

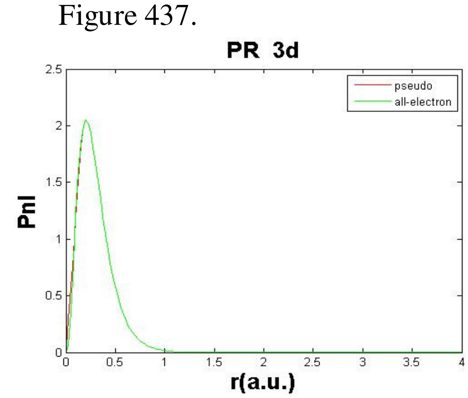


Figure 439.

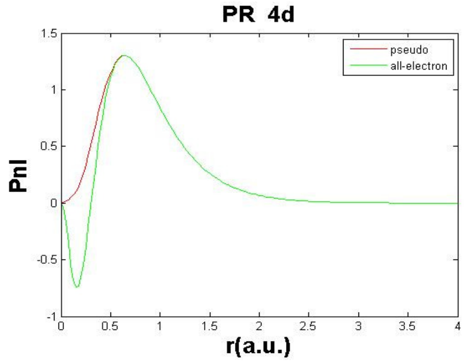


Figure 440.

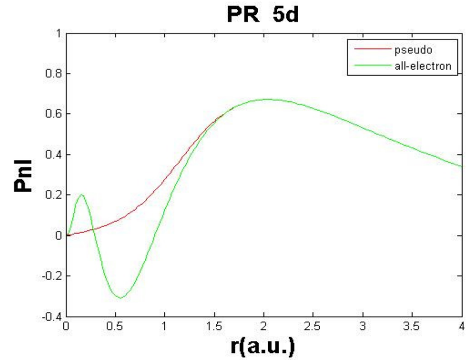


Figure 441.

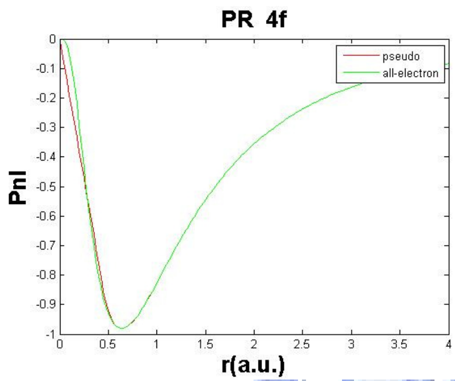


Figure 442.

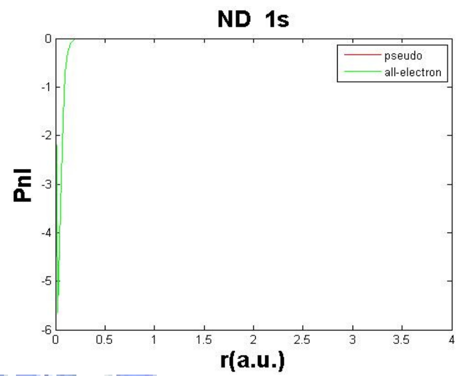


Figure 443.

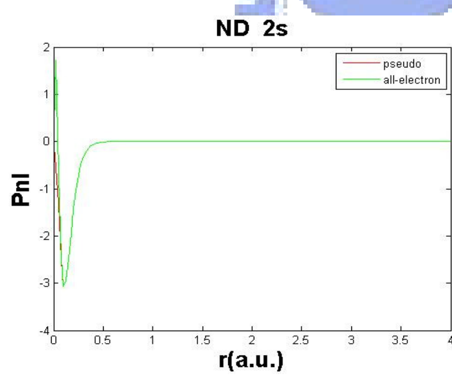


Figure 444.

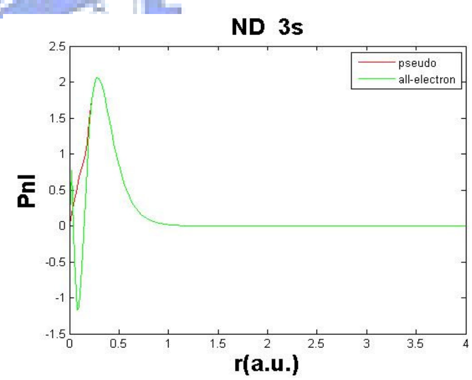


Figure 445.

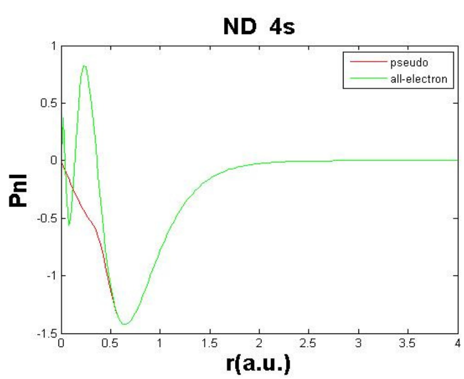


Figure 446.

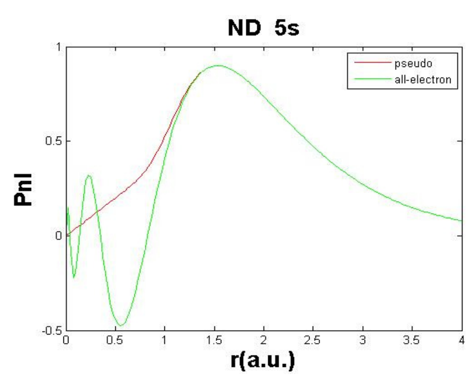


Figure 447.

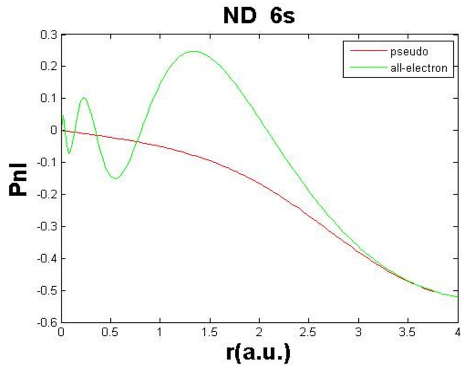


Figure 448.

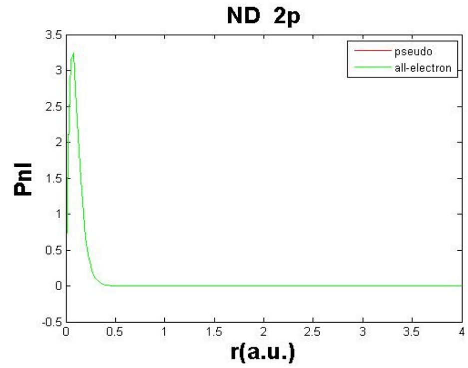


Figure 449.

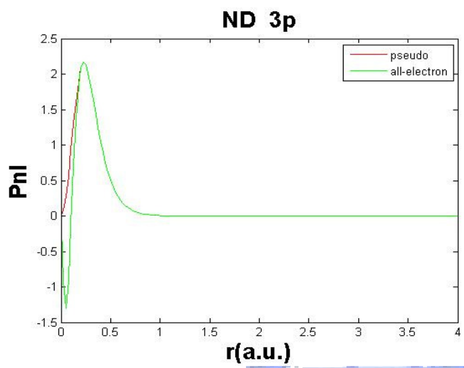


Figure 450.

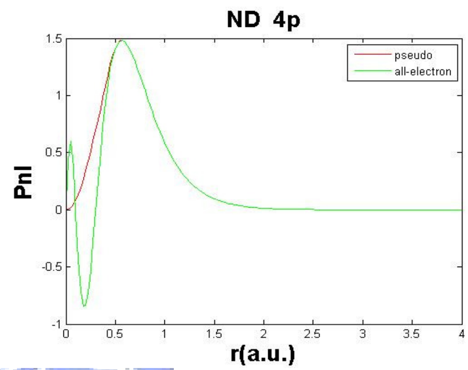


Figure 451.

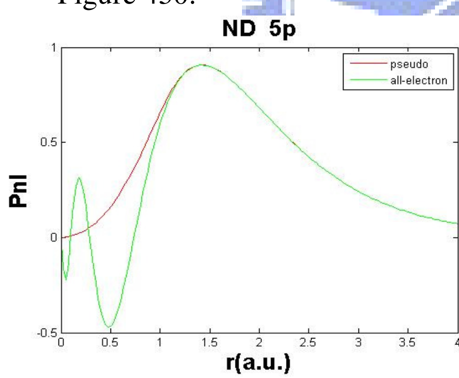


Figure 452.

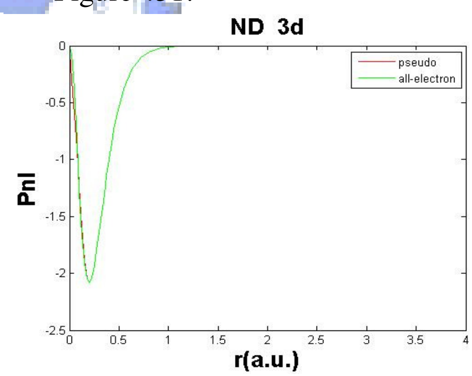


Figure 453.

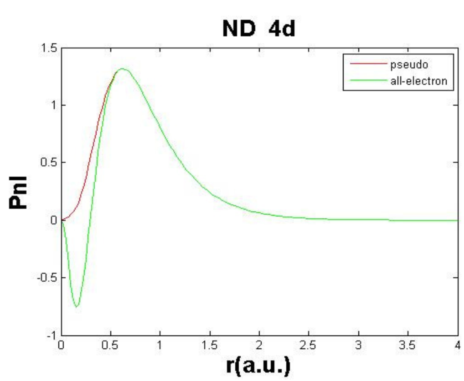


Figure 454.

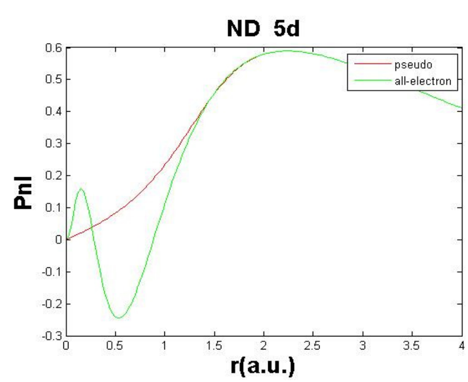


Figure 455.

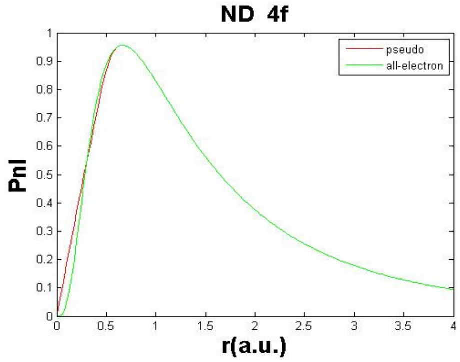


Figure 456.

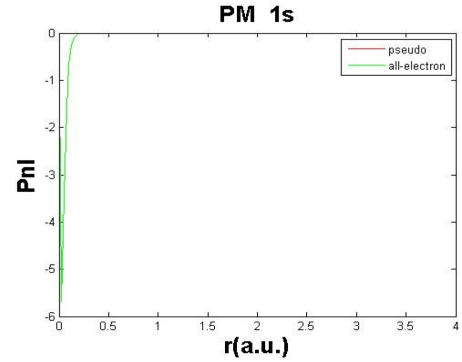


Figure 457.

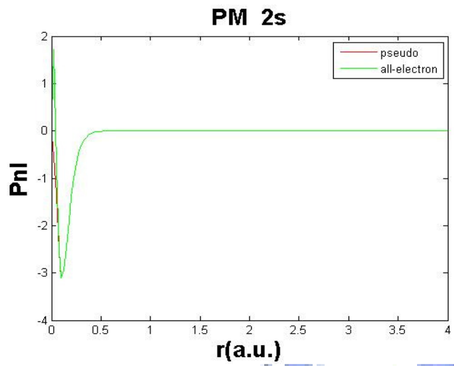


Figure 458.

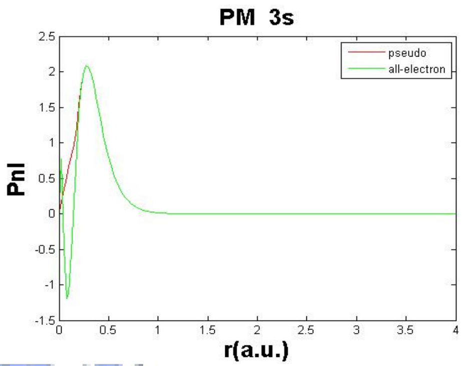


Figure 459.

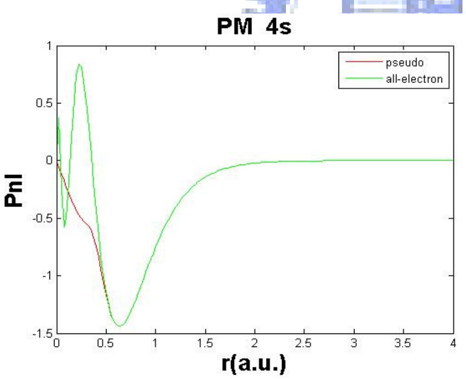


Figure 460.

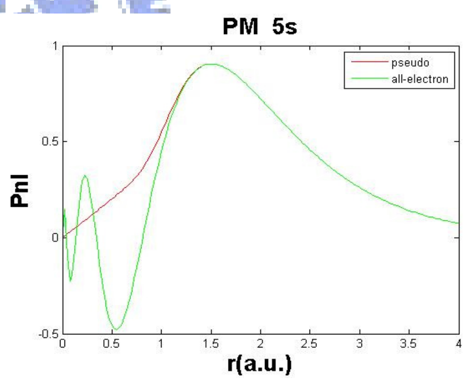


Figure 461.

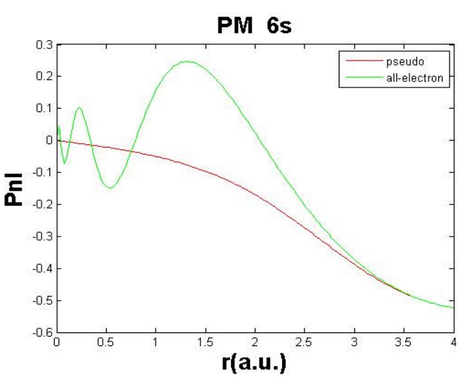


Figure 462.

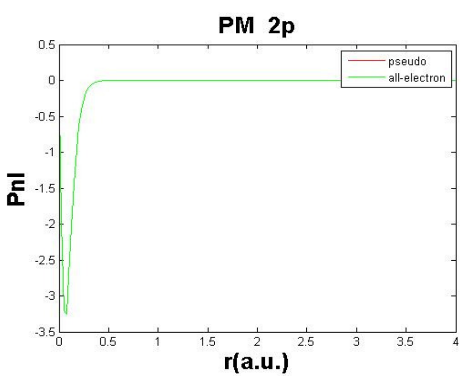


Figure 463.

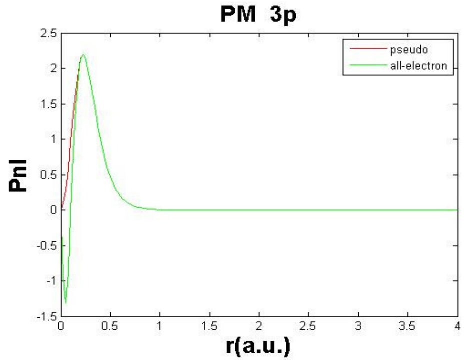


Figure 464.

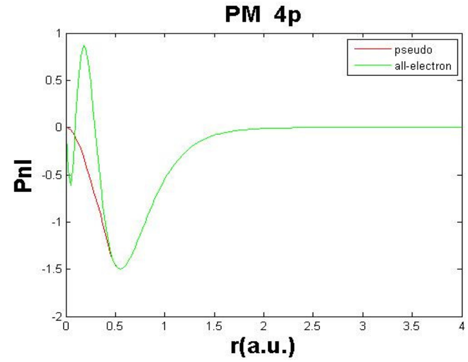


Figure 465.

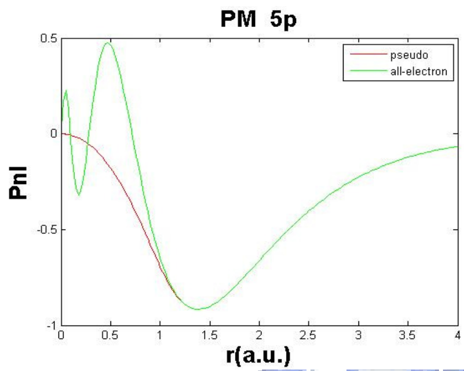


Figure 466.

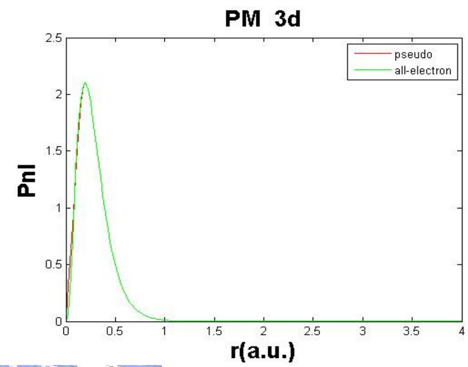


Figure 467.

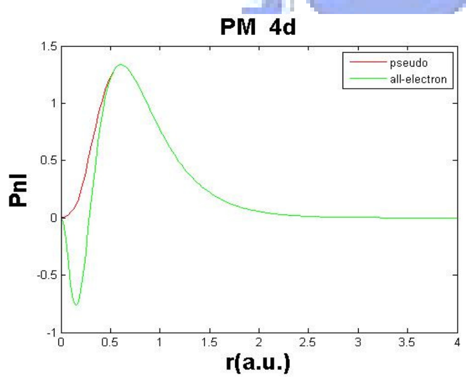


Figure 468.

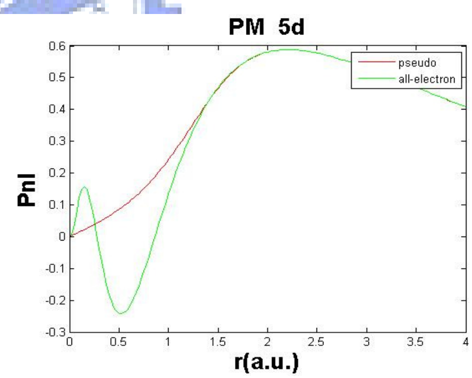


Figure 469.

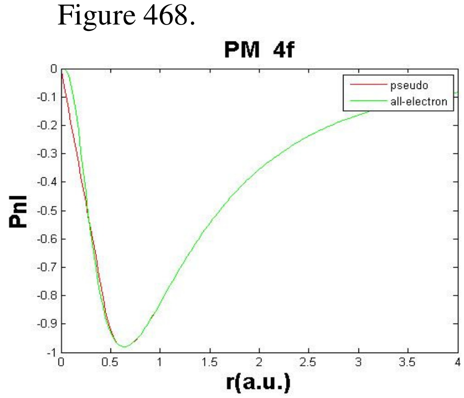


Figure 470.

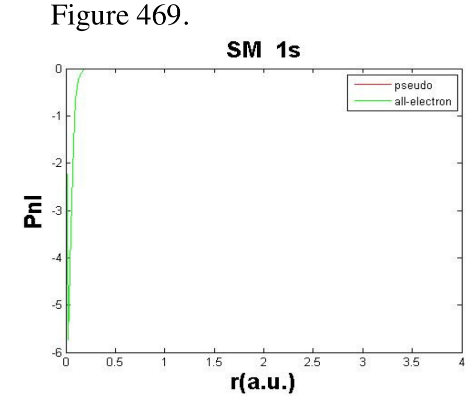


Figure 471.

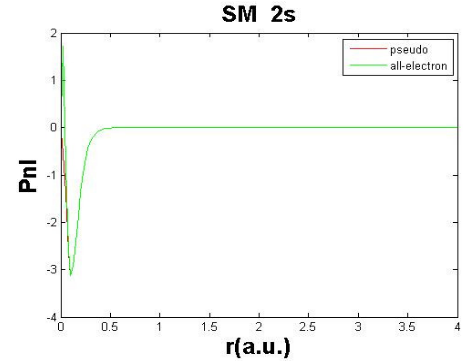


Figure 472.

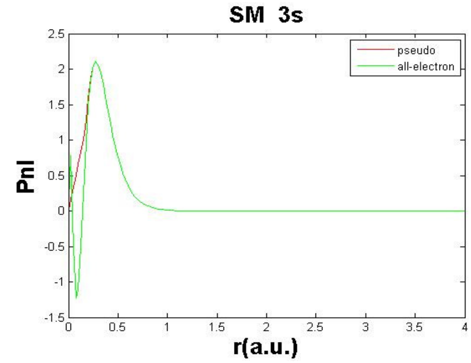


Figure 473.

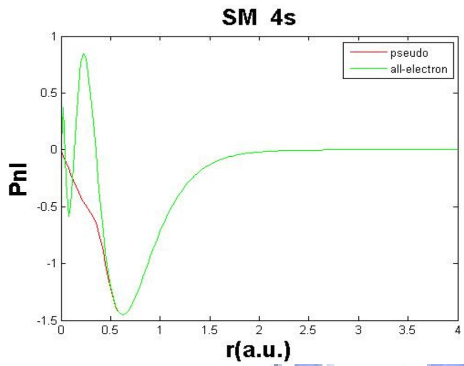


Figure 474.

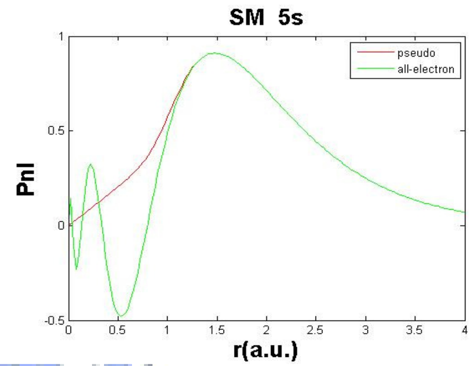


Figure 475.

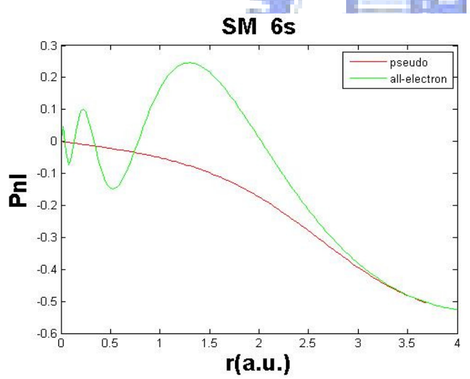


Figure 476.

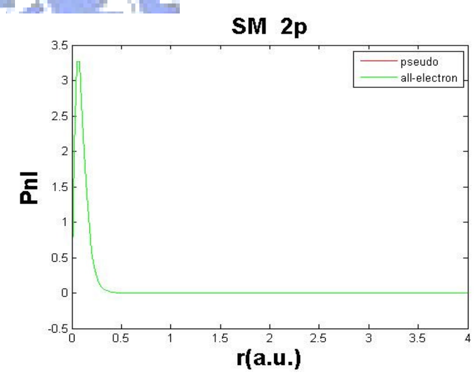


Figure 477.

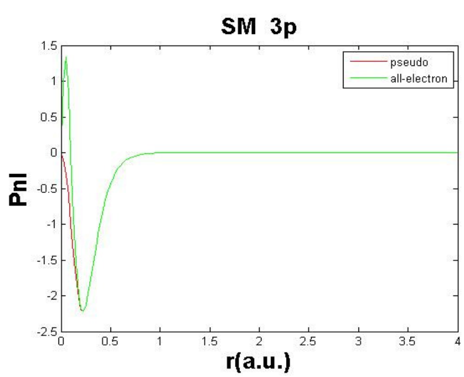


Figure 478.

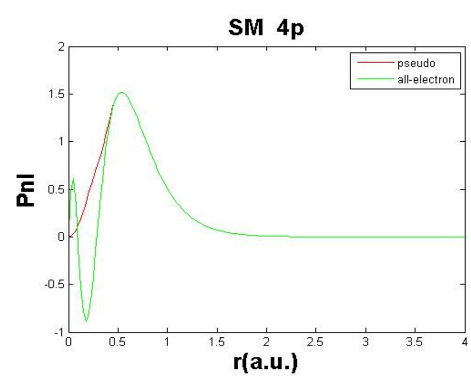


Figure 479.

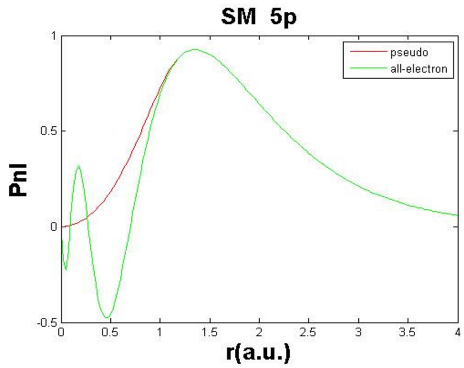


Figure 480.

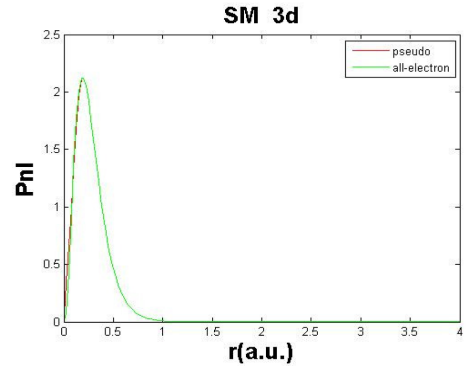


Figure 481.

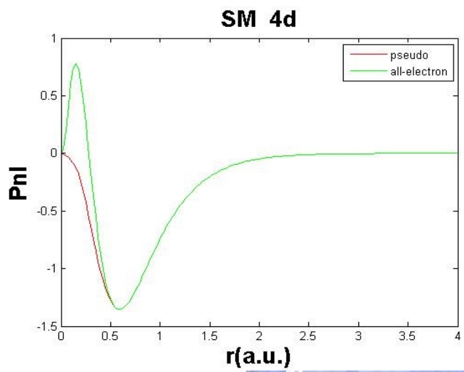


Figure 482.

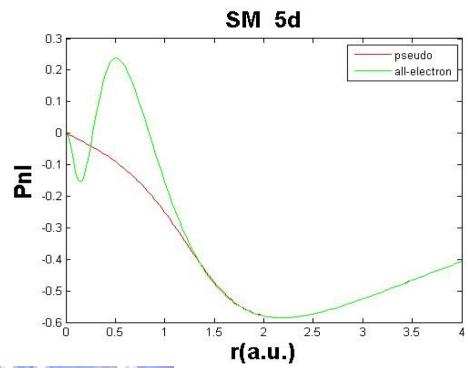


Figure 483.

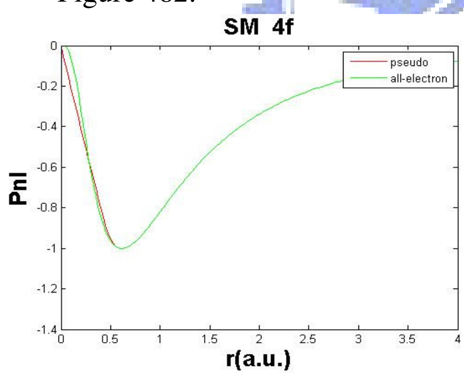


Figure 484.

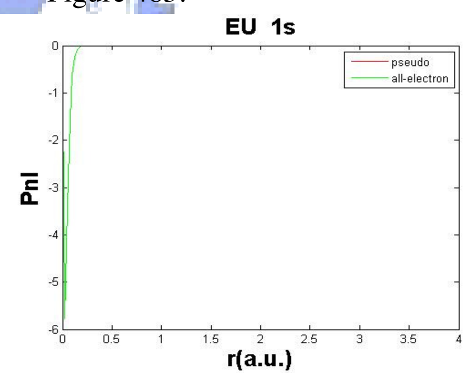


Figure 485.

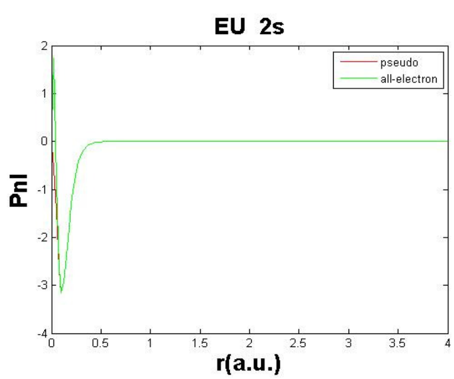


Figure 486.

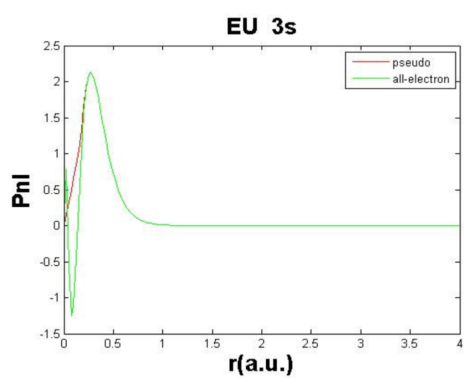


Figure 487.

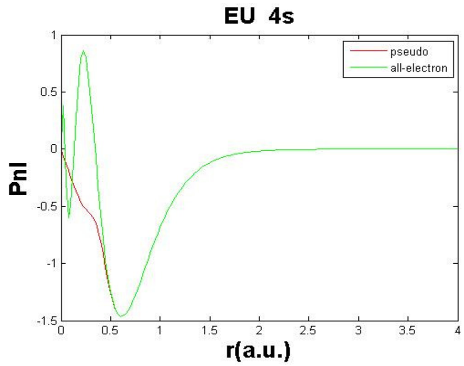


Figure 488.

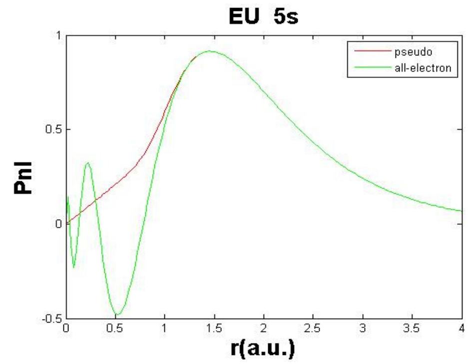


Figure 489.

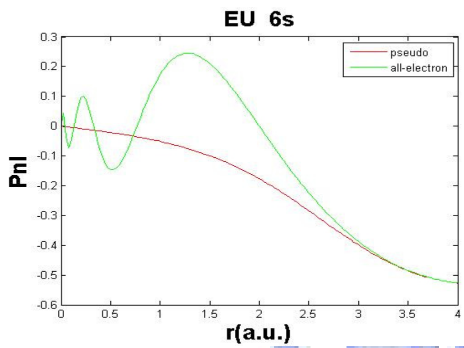


Figure 490.

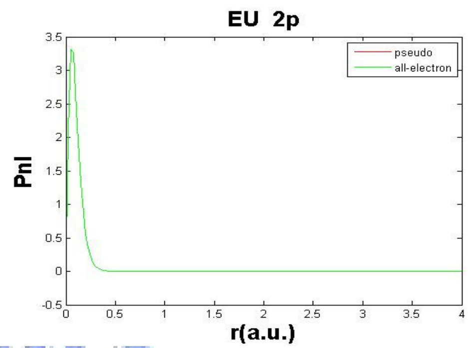


Figure 491.

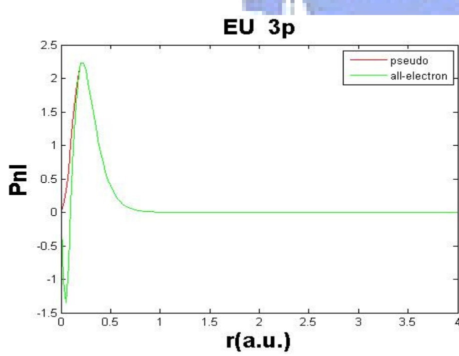


Figure 492.

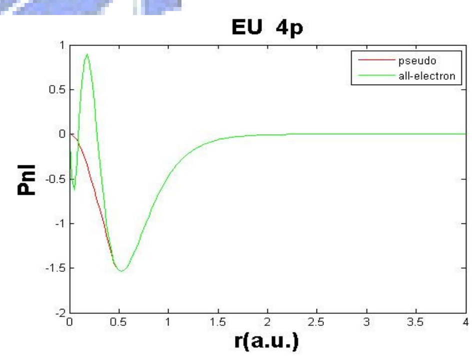


Figure 493.

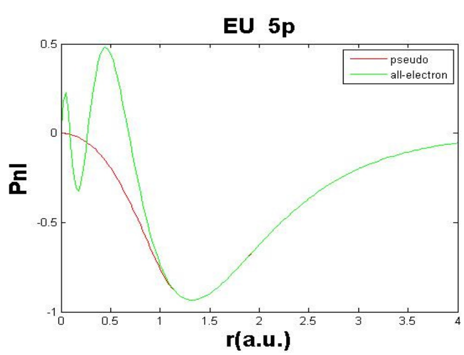


Figure 494.

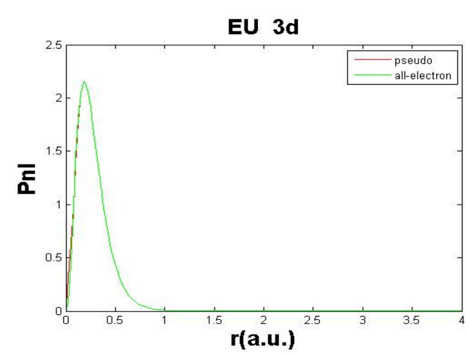


Figure 495.

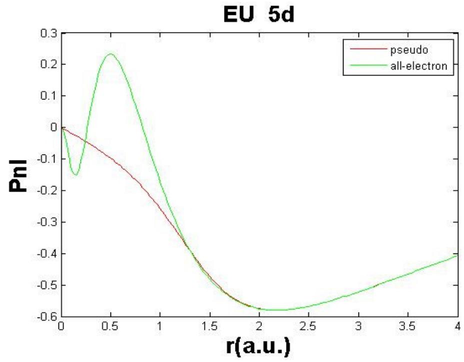


Figure 496.

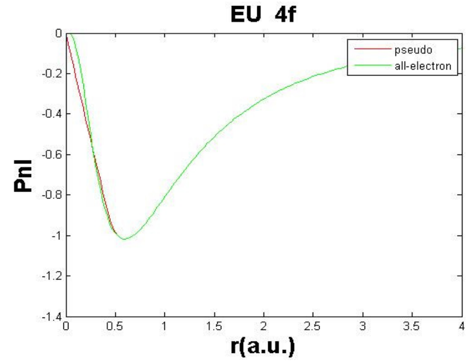


Figure 497.

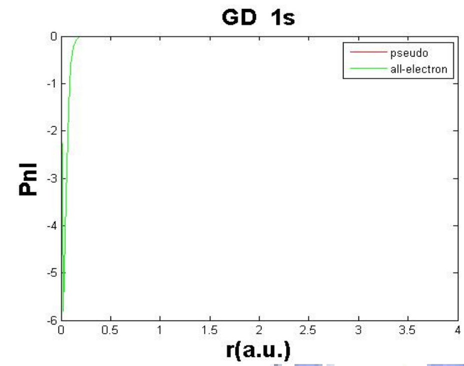


Figure 498.

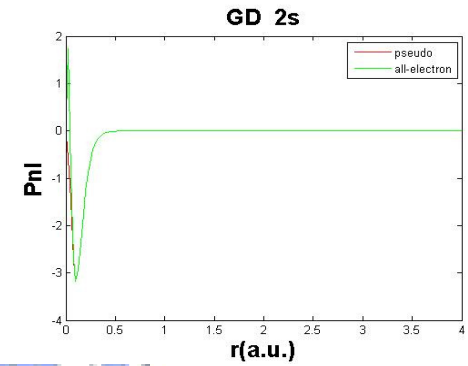


Figure 499.

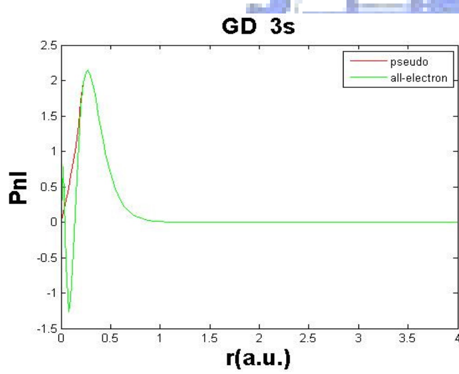


Figure 500.

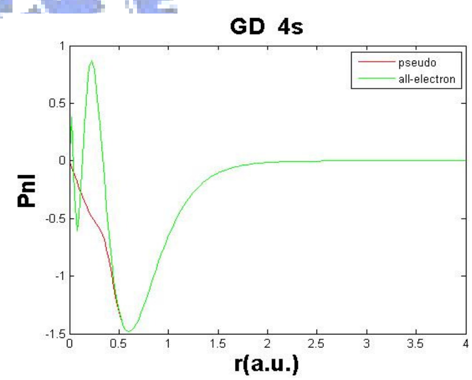


Figure 501.

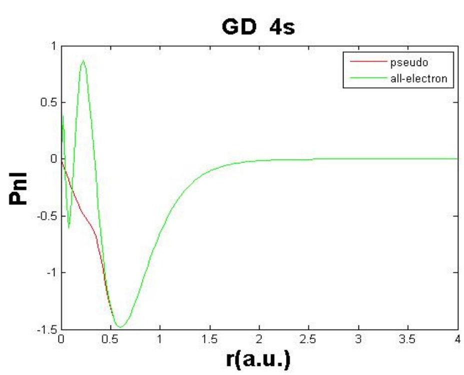


Figure 502.

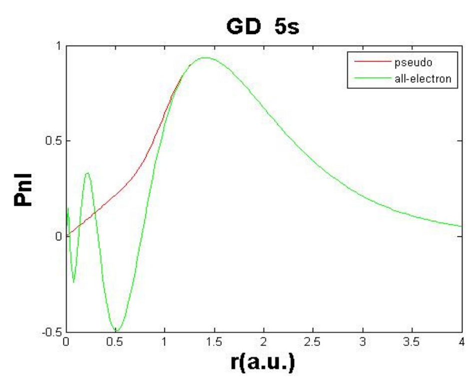


Figure 503.

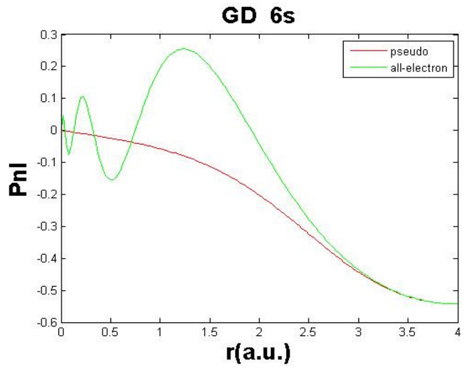


Figure 504.

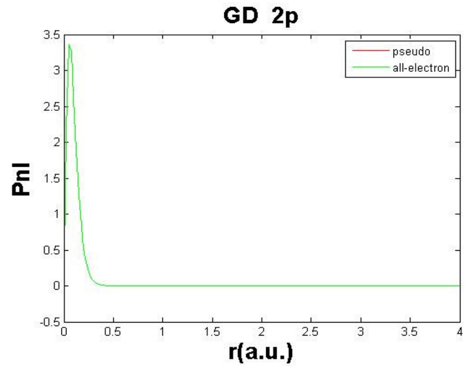


Figure 505.

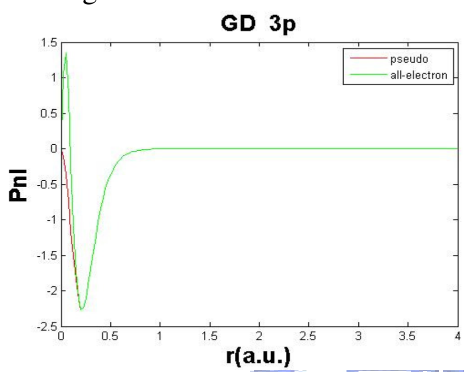


Figure 506.

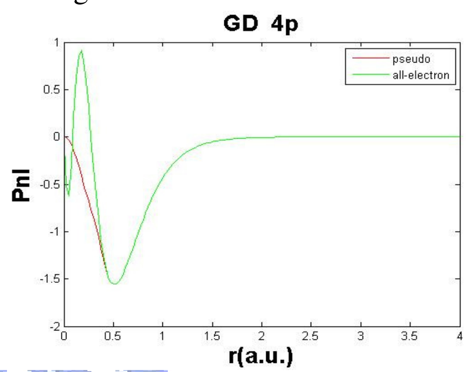


Figure 507.

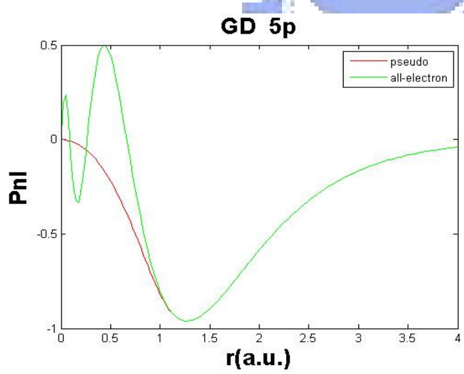


Figure 508.

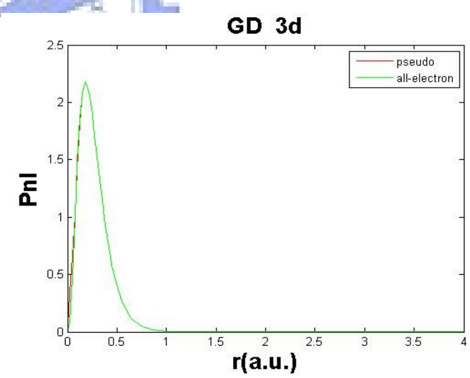


Figure 509.

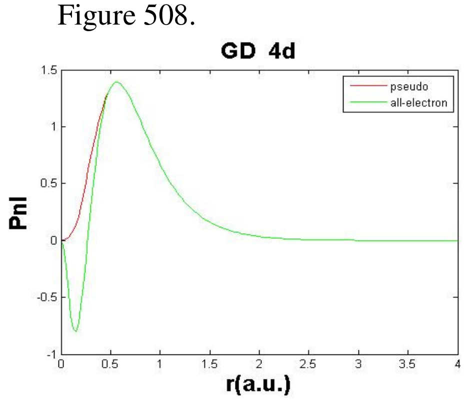


Figure 510.

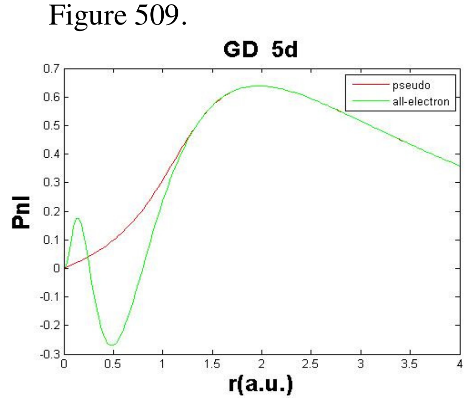


Figure 511.

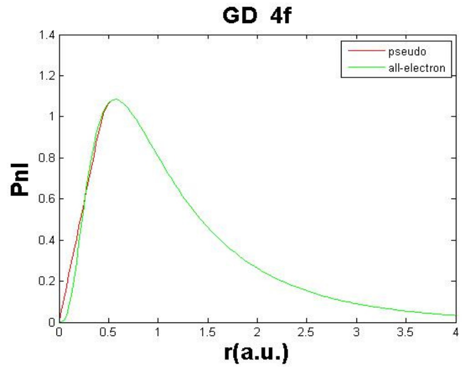


Figure 512.

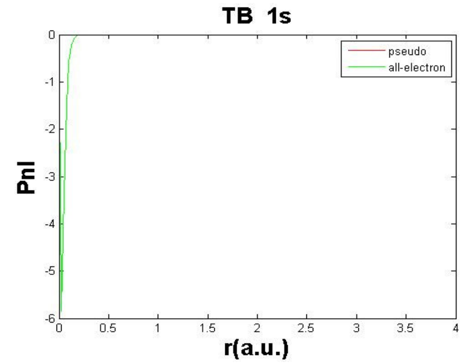


Figure 513.

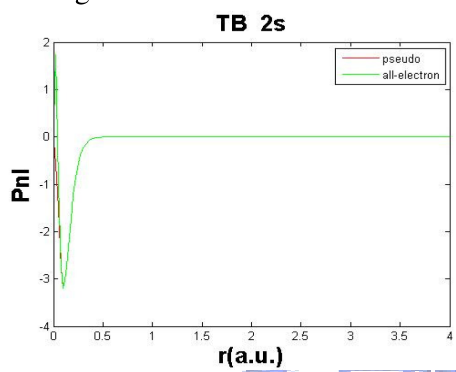


Figure 514.

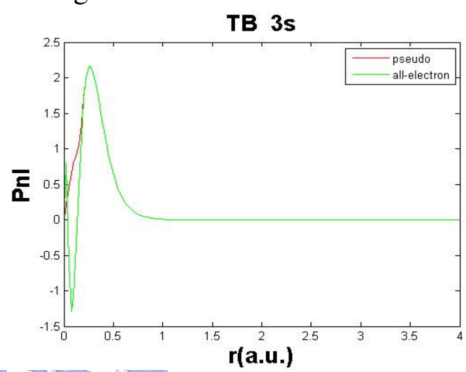


Figure 515.

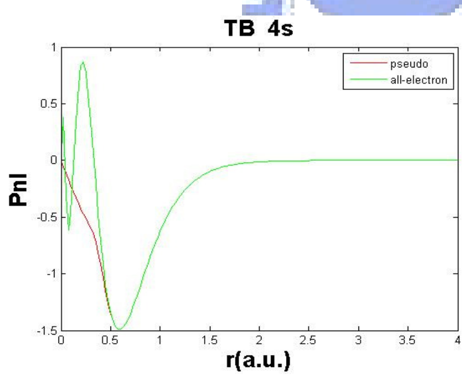


Figure 516.

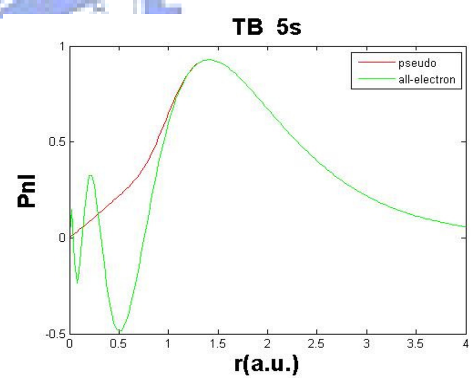


Figure 517.

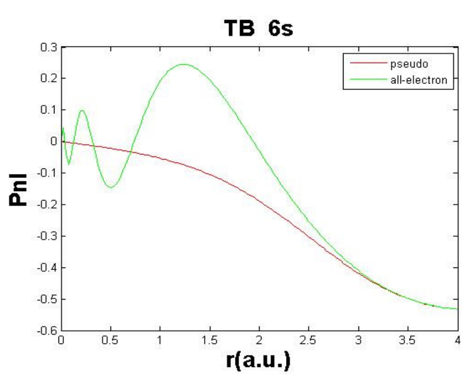


Figure 518.

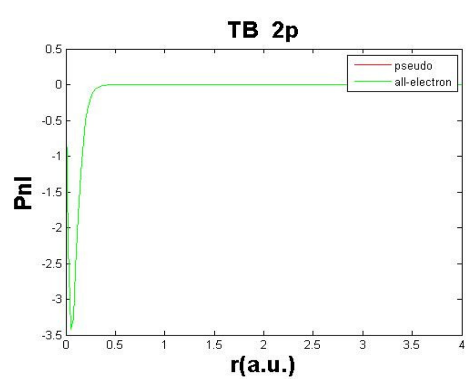


Figure 519.

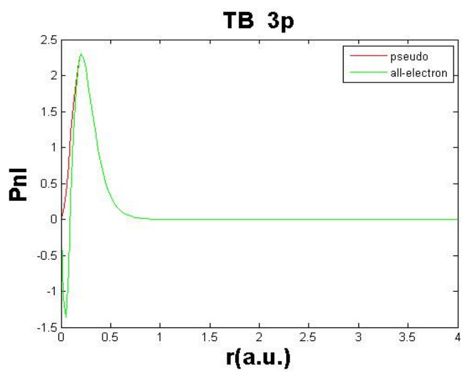


Figure 520.

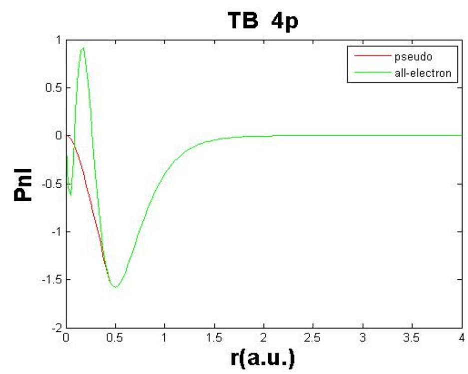


Figure 521.

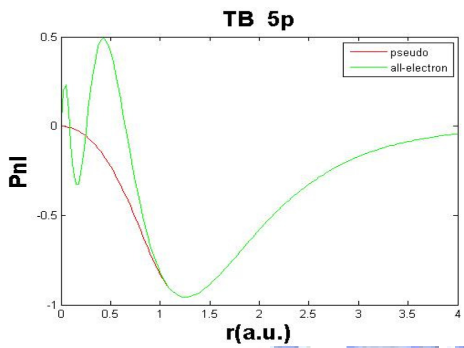


Figure 522.

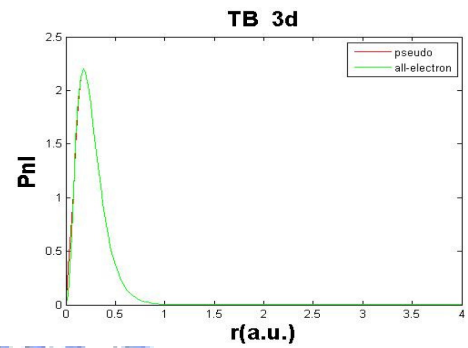


Figure 523.

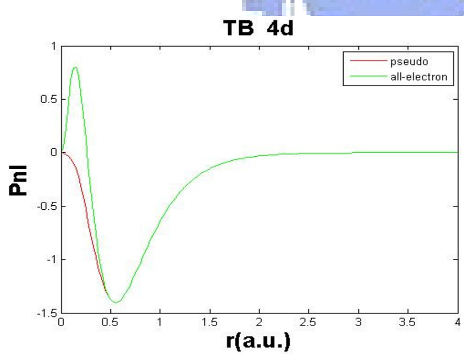


Figure 524.

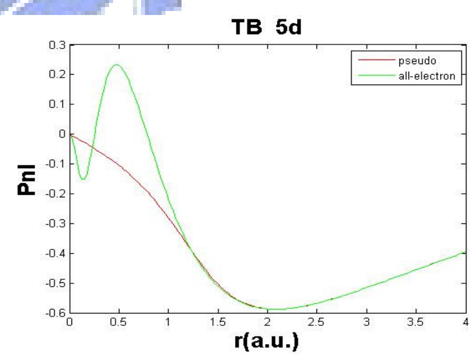


Figure 525.

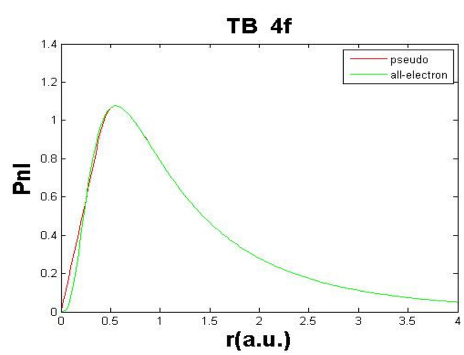


Figure 526.

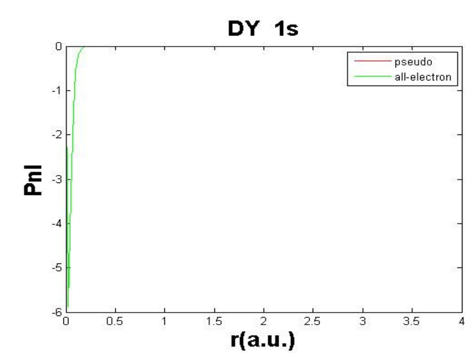


Figure 527.

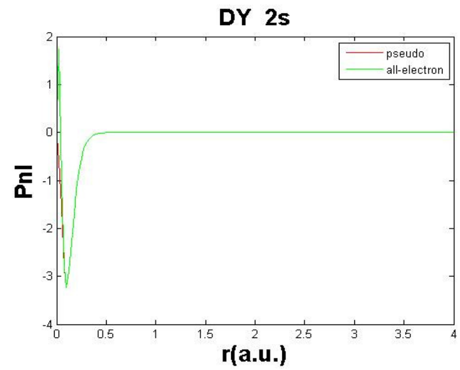


Figure 528.

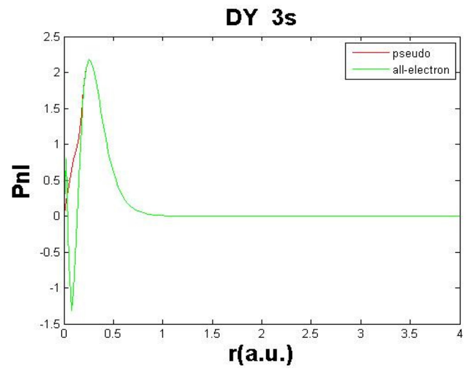


Figure 529.

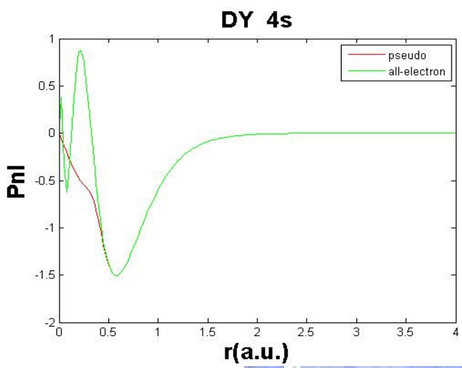


Figure 530.

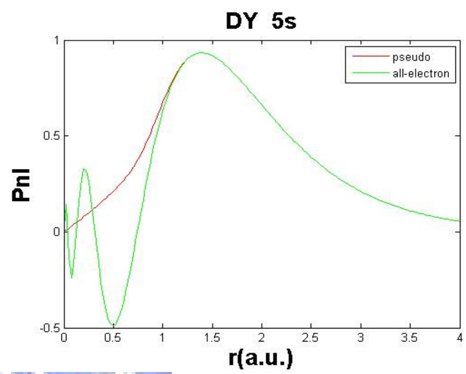


Figure 531.

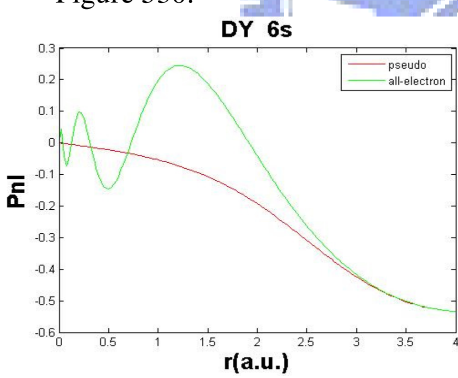


Figure 532.

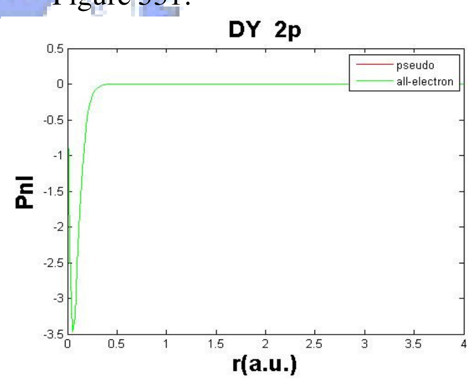


Figure 533.

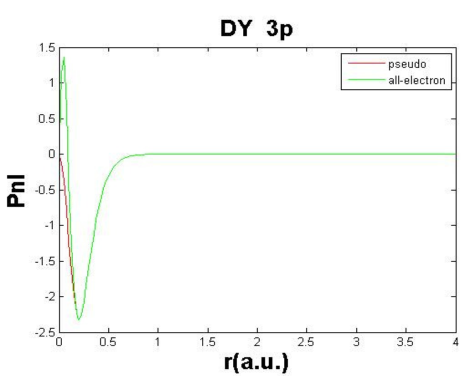


Figure 534.

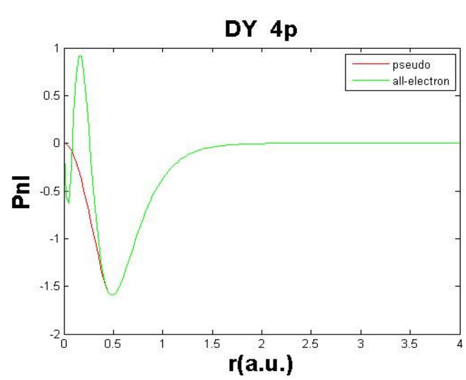


Figure 535.

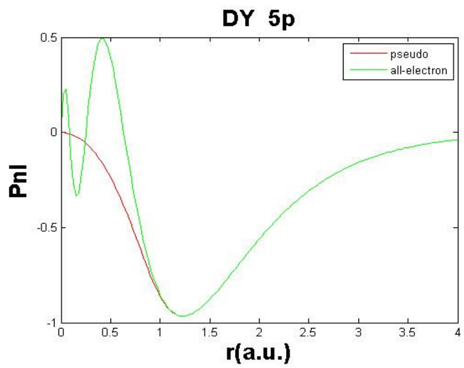


Figure 536.

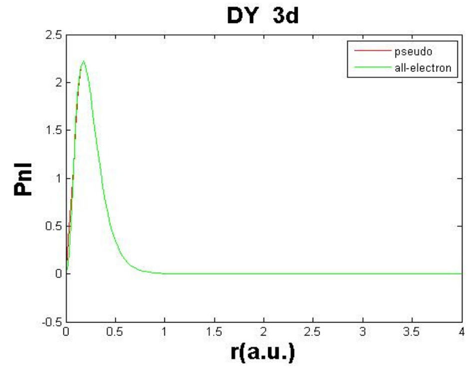


Figure 537.

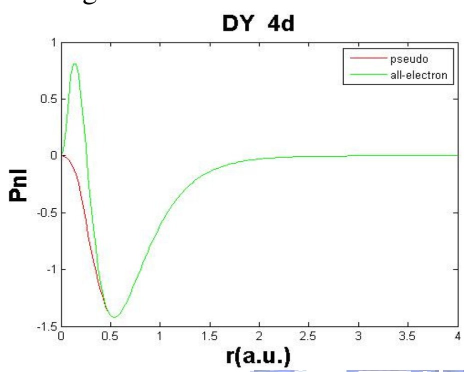


Figure 538.

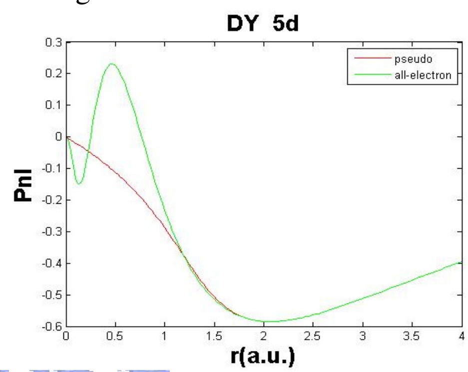


Figure 539.

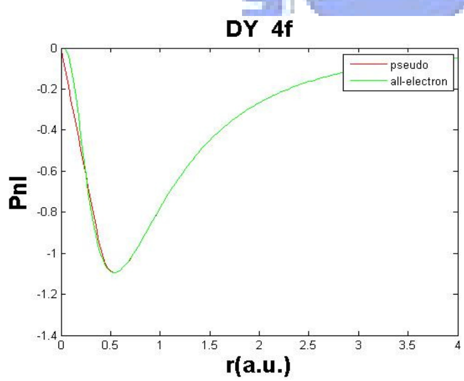


Figure 540.

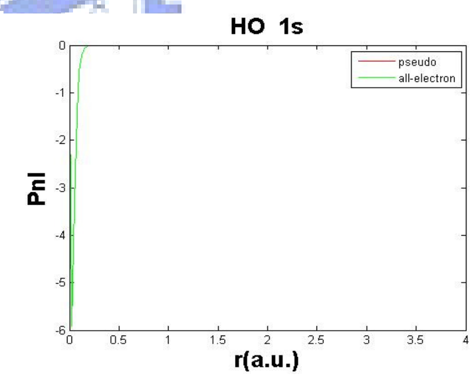


Figure 541.

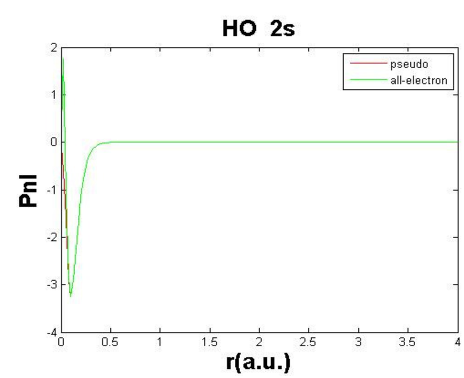


Figure 542.

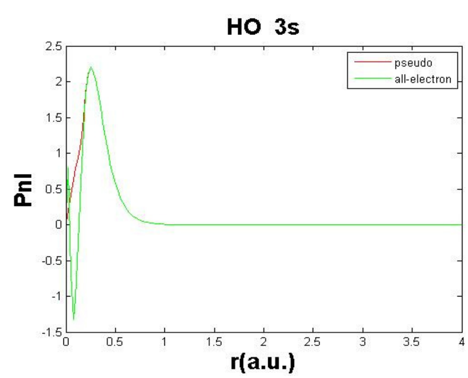


Figure 543.

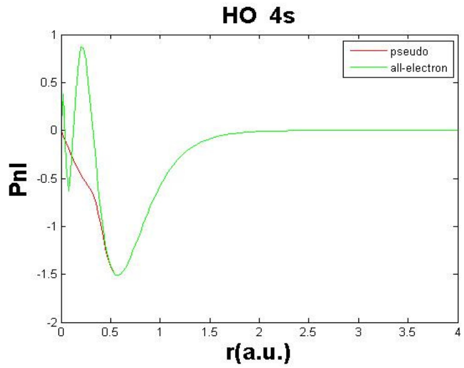


Figure 544.

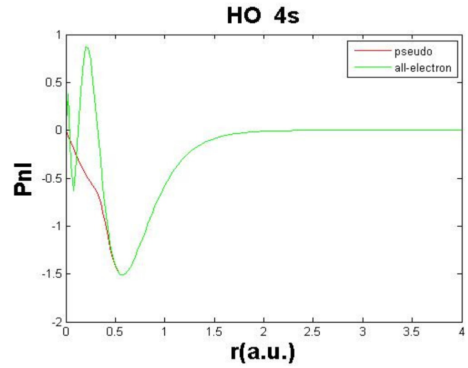


Figure 545.

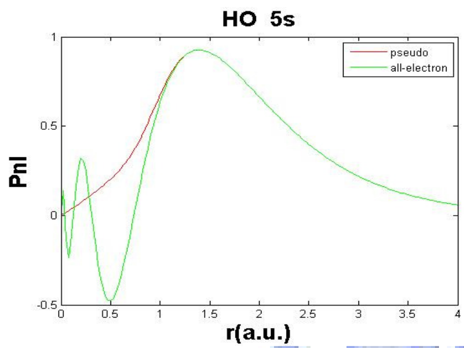


Figure 546.

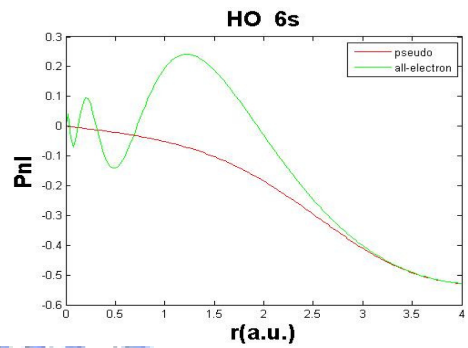


Figure 547.

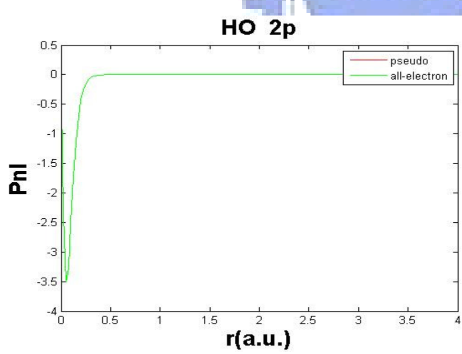


Figure 548.

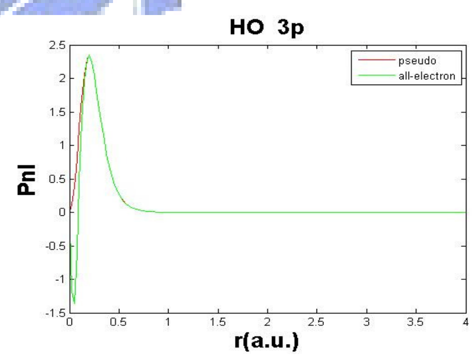


Figure 549.

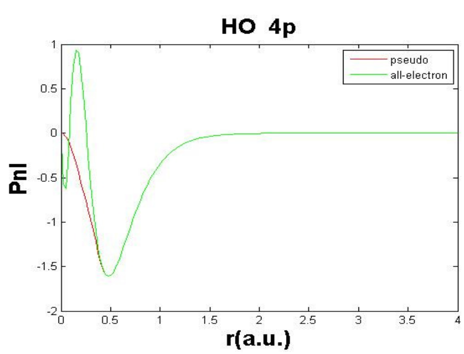


Figure 550.

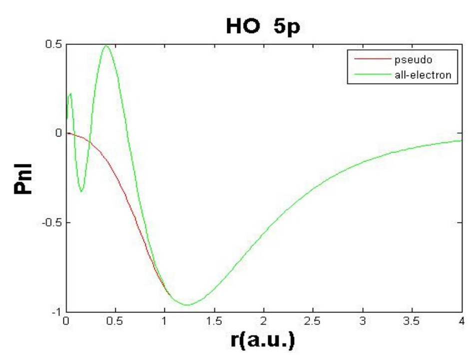


Figure 551.

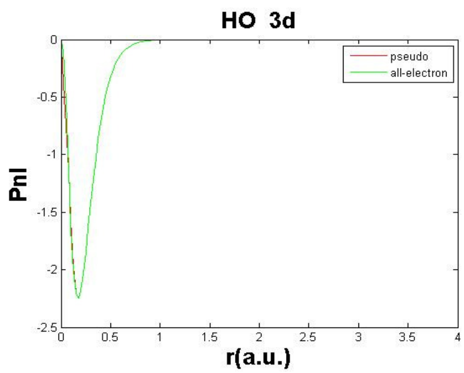


Figure 552.

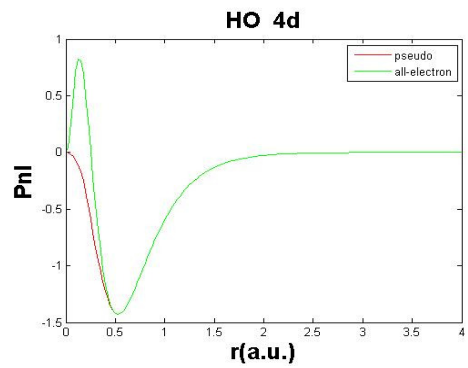


Figure 553.

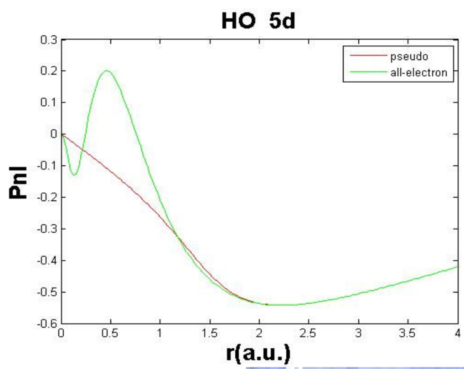


Figure 554.

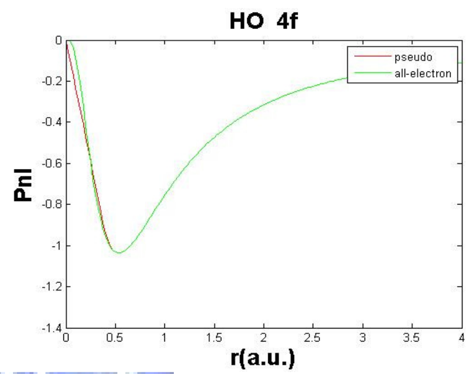


Figure 555.

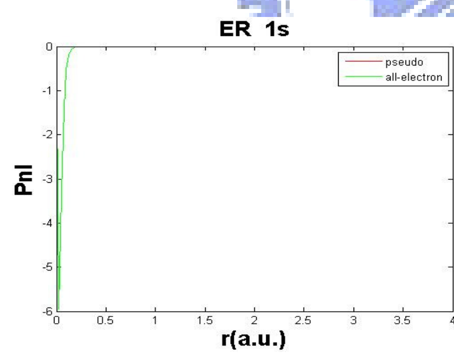


Figure 556.

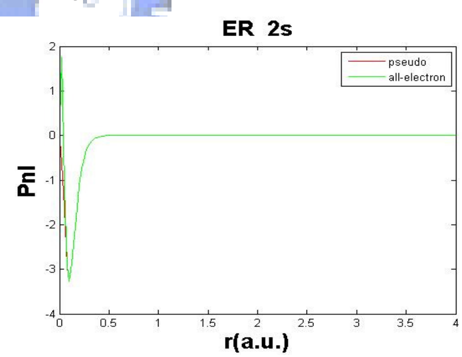


Figure 557.

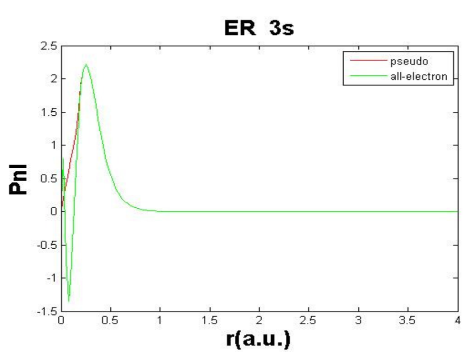


Figure 558.

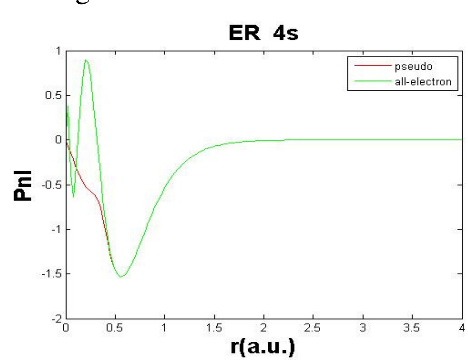


Figure 559.

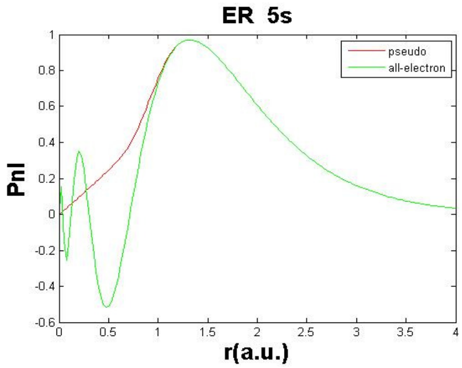


Figure 560.

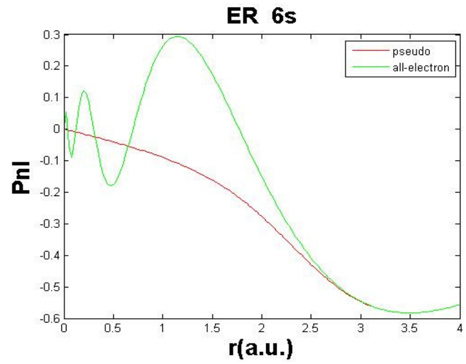


Figure 561.

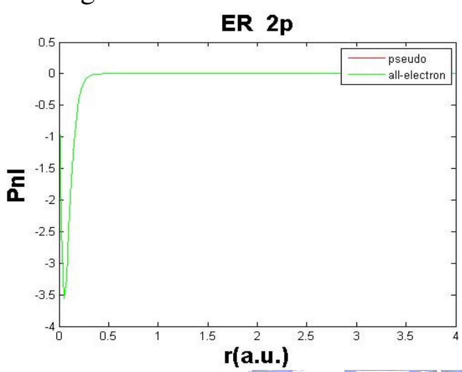


Figure 562.

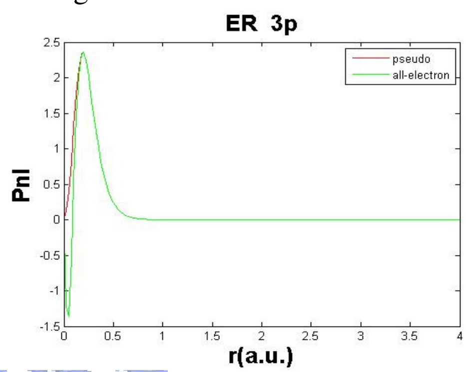


Figure 563.

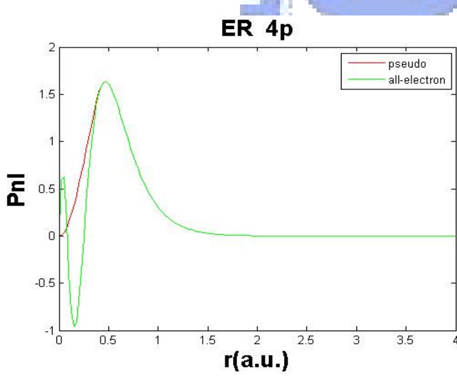


Figure 564.

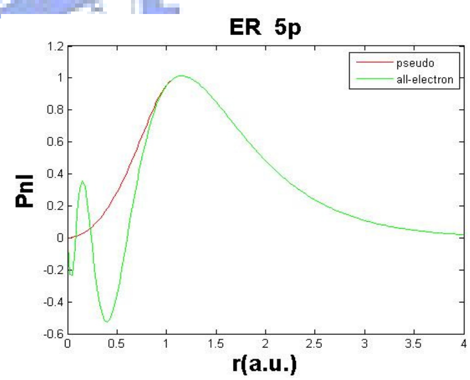


Figure 565.

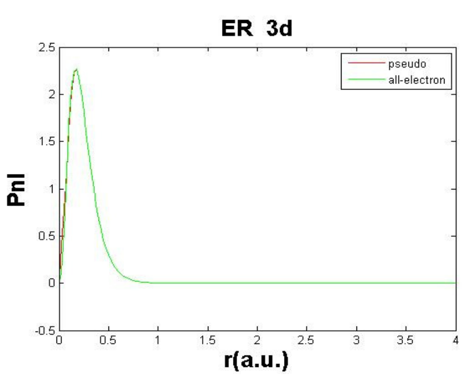


Figure 566.

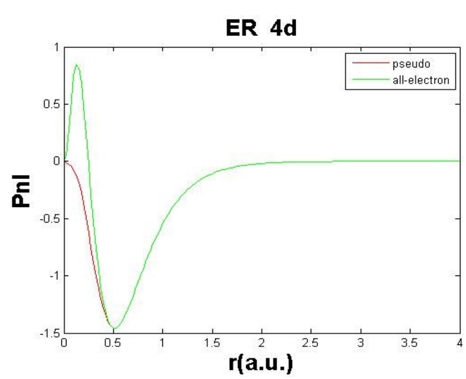


Figure 567.

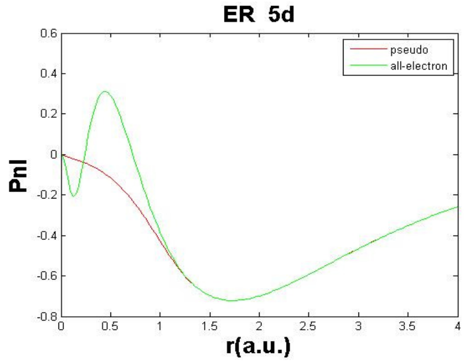


Figure 568.

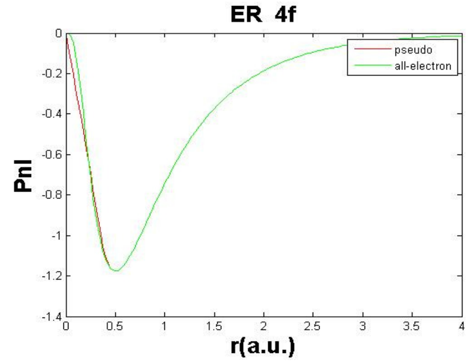


Figure 569.

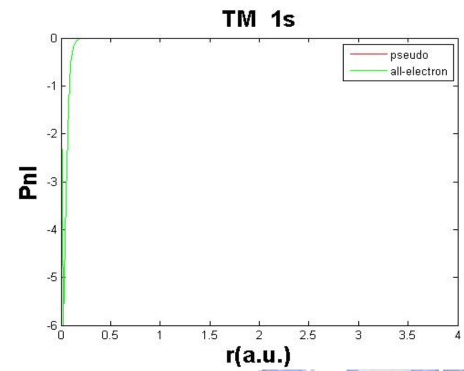


Figure 570.

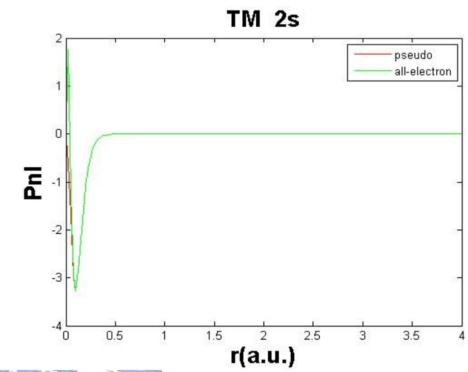


Figure 571.

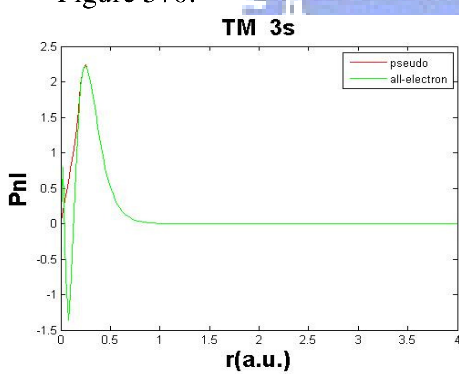


Figure 572.

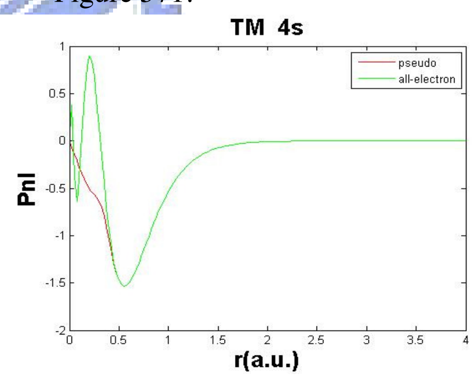


Figure 573.

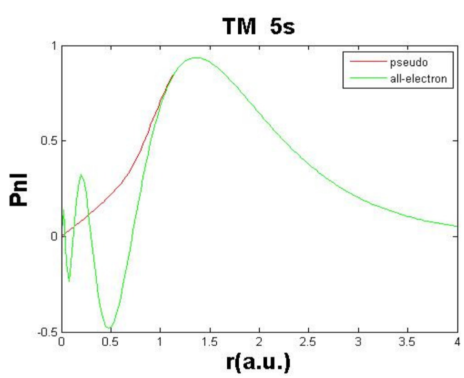


Figure 574.

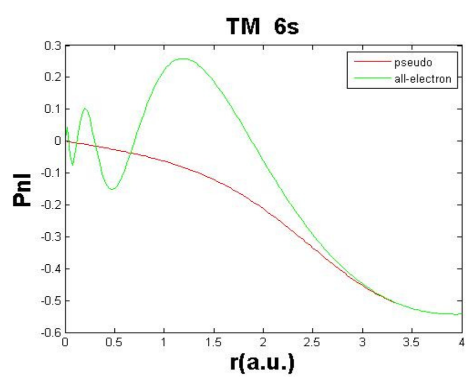


Figure 575.

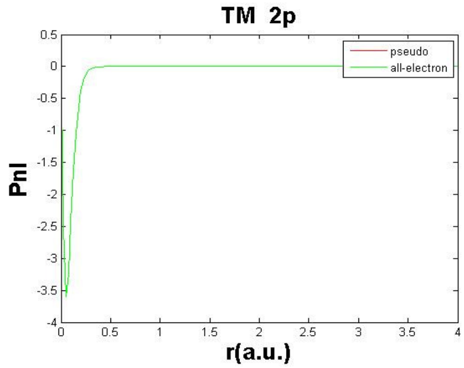


Figure 576.

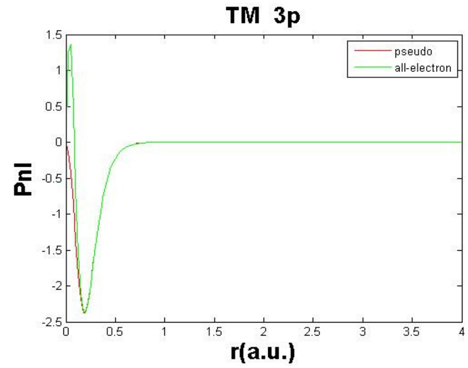


Figure 577.

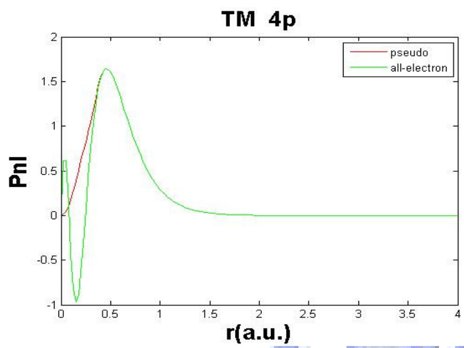


Figure 578.

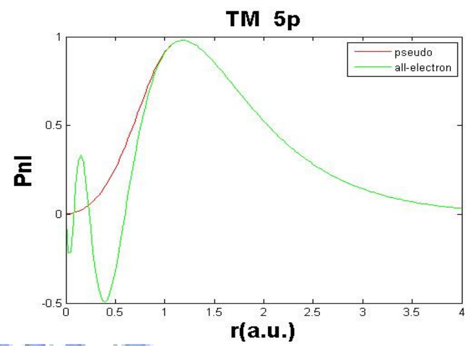


Figure 579.

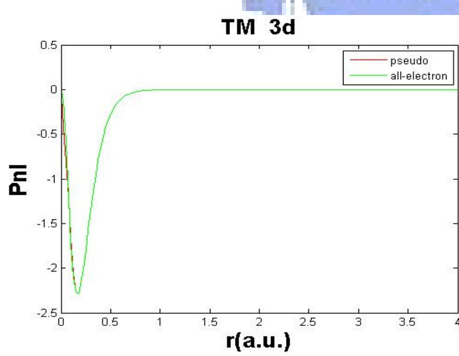


Figure 580.

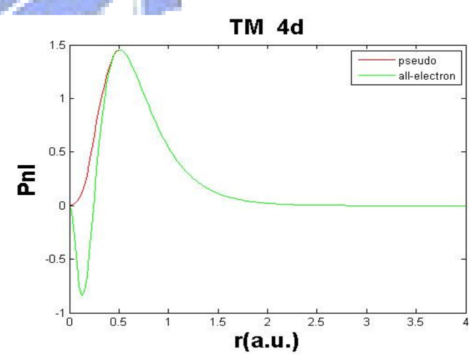


Figure 581.

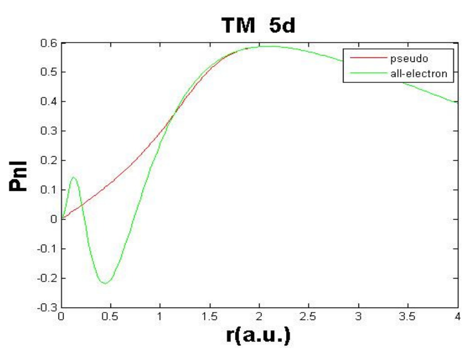


Figure 582.

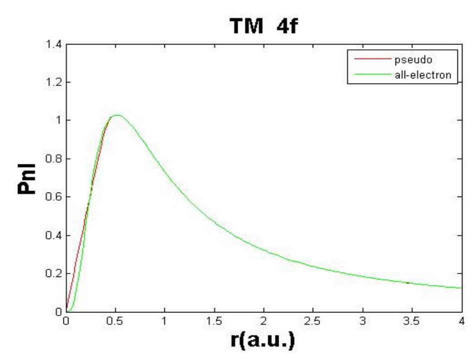


Figure 583.

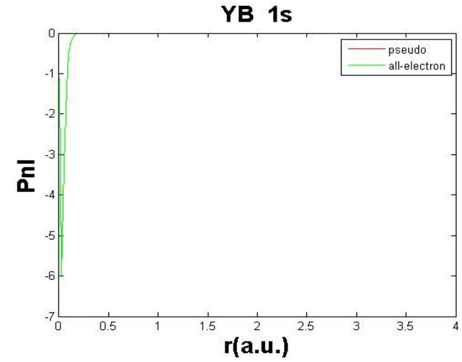


Figure 584.

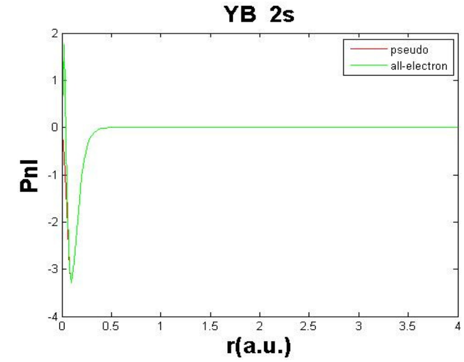


Figure 585.

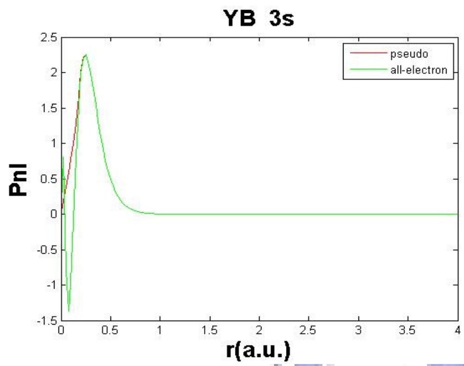


Figure 586.

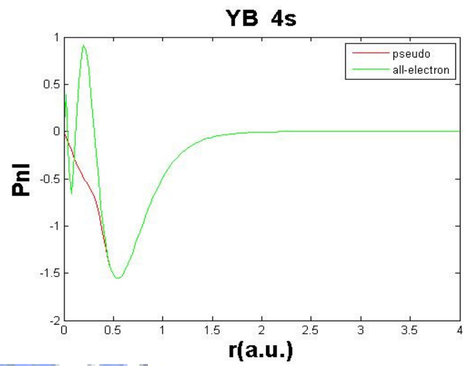


Figure 587.

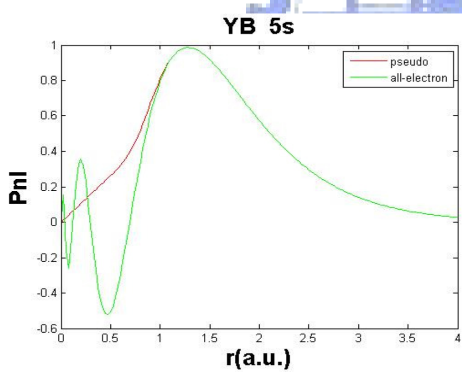


Figure 588.

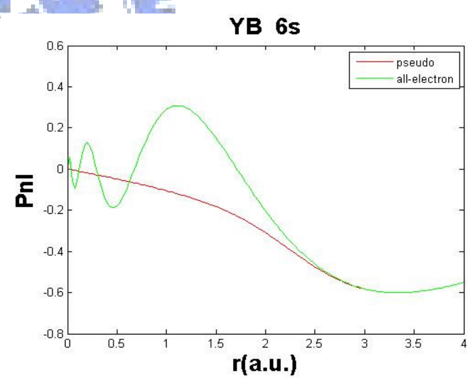


Figure 589.

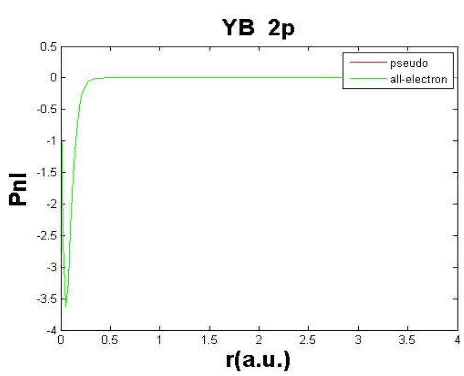


Figure 590.

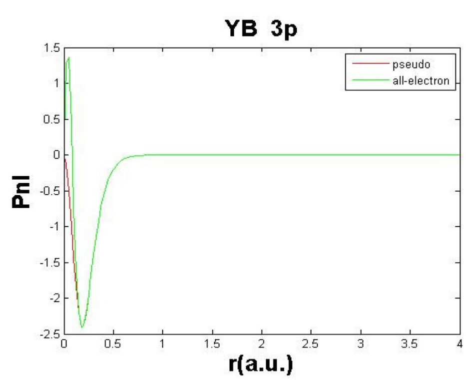


Figure 591.

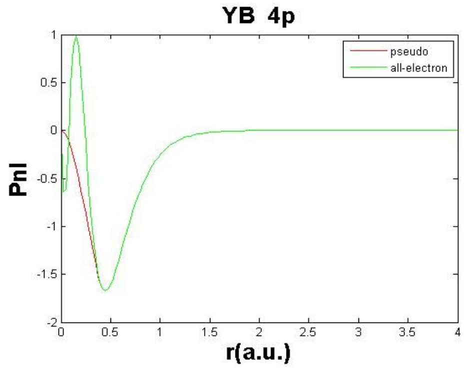


Figure 592.

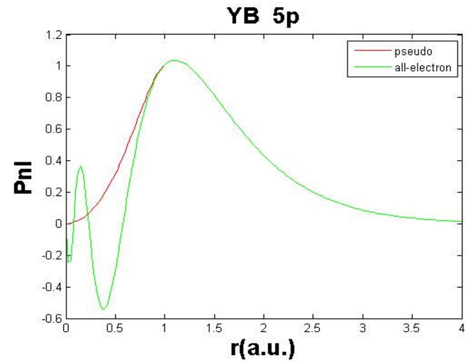


Figure 593.

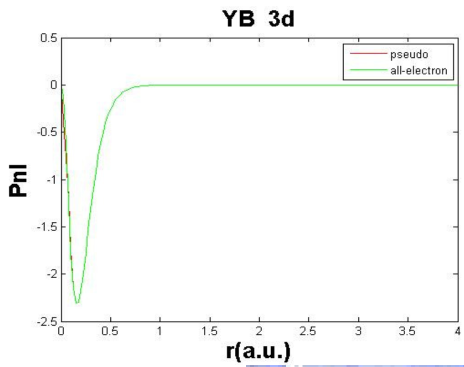


Figure 594.

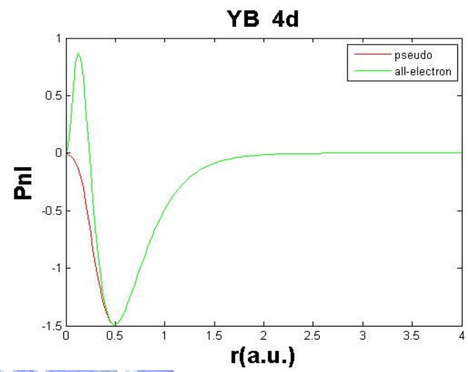


Figure 595.

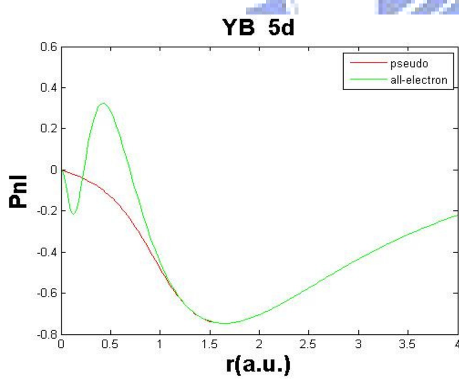


Figure 596.

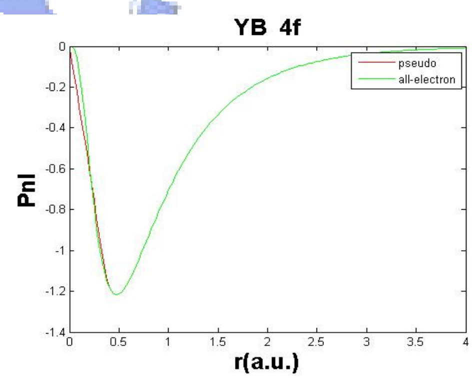


Figure 597.

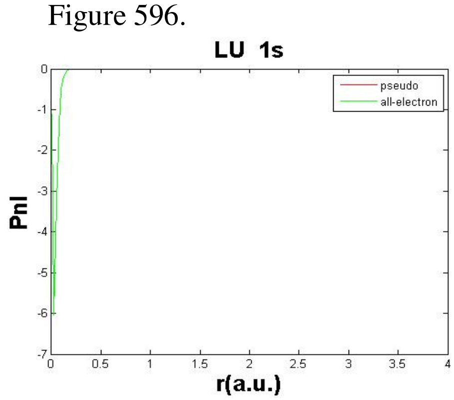


Figure 598.

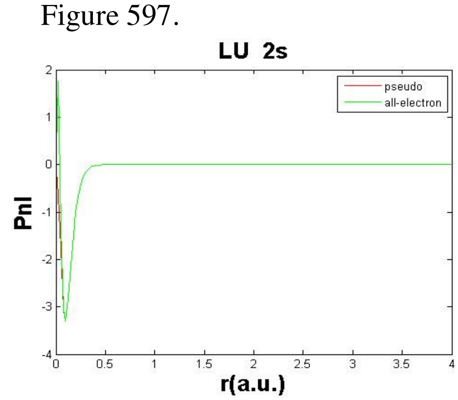


Figure 599.

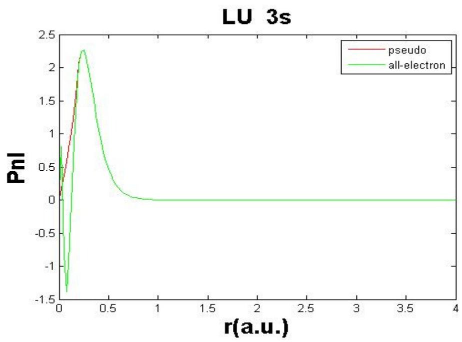


Figure 600.

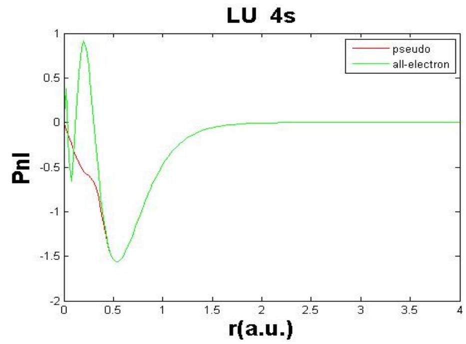


Figure 601.

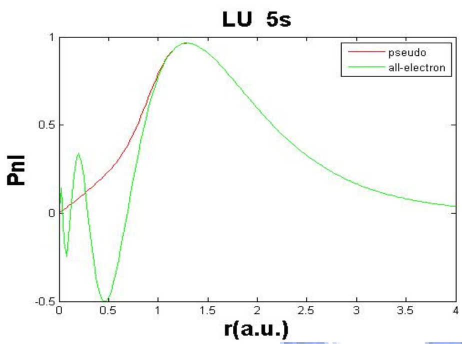


Figure 602.

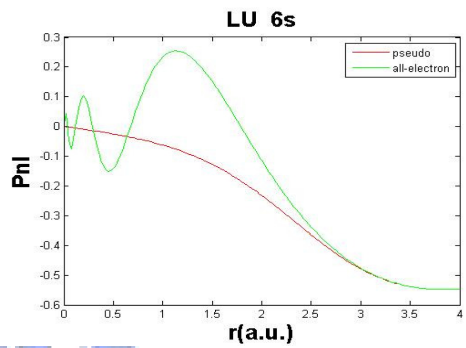


Figure 603.

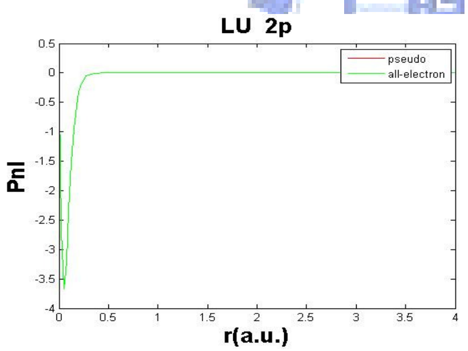


Figure 604.

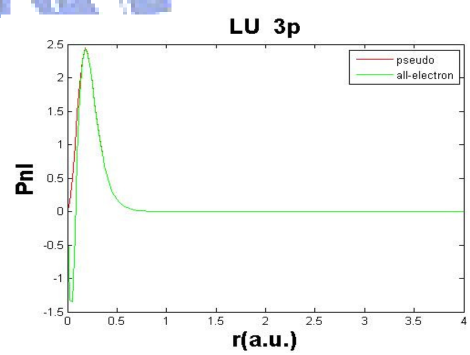


Figure 605.

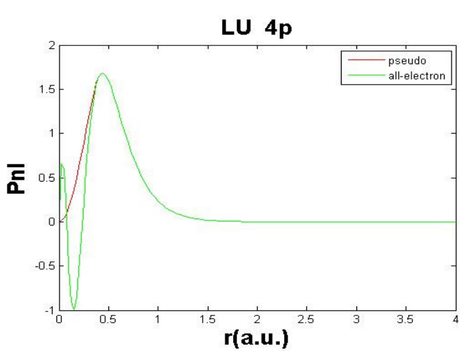


Figure 606.

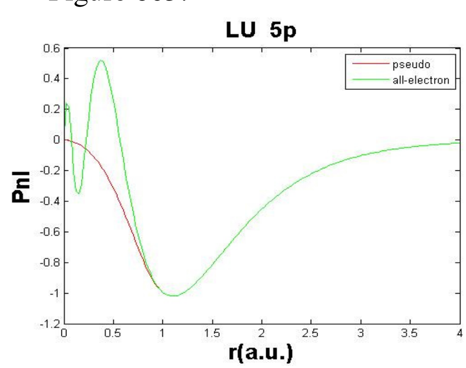


Figure 607.

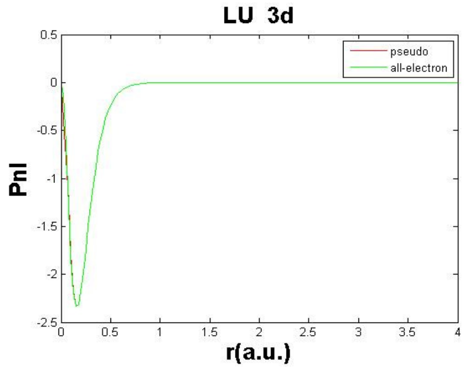


Figure 608.

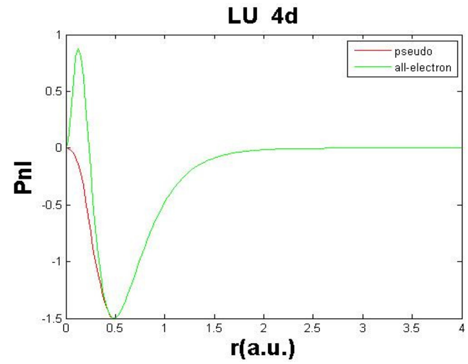


Figure 609.

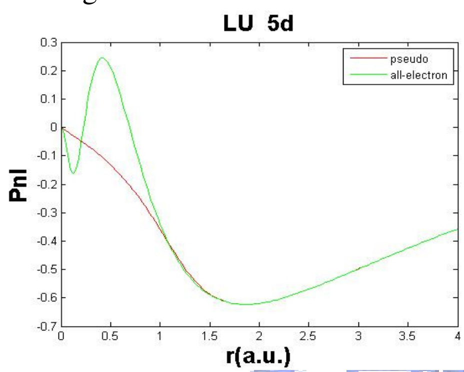


Figure 610.

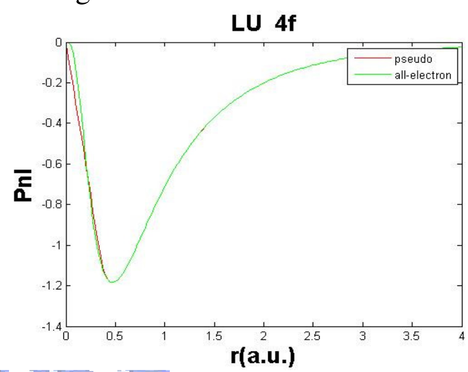


Figure 611.

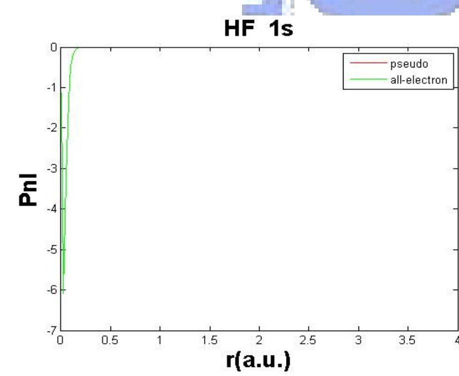


Figure 612.

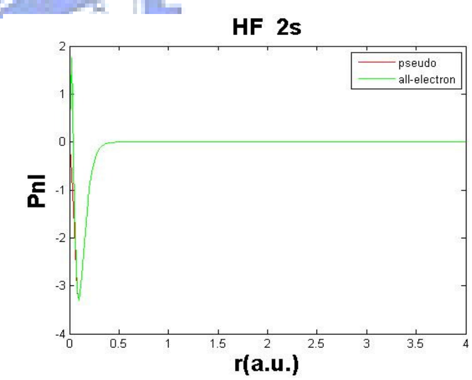


Figure 613.

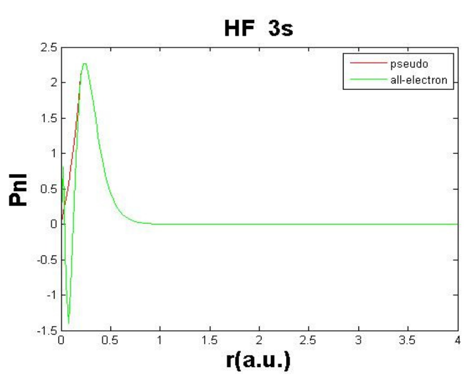


Figure 614.

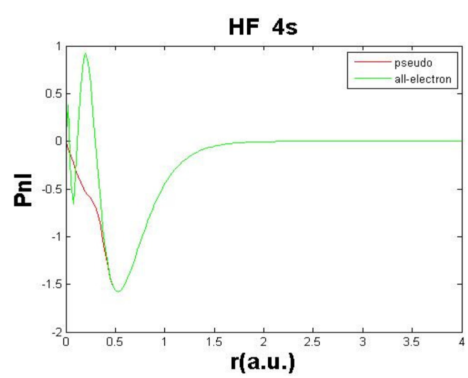


Figure 615.

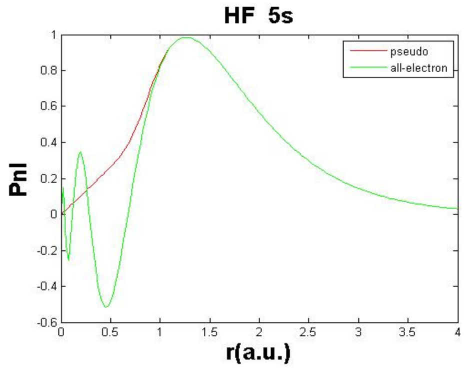


Figure 616.

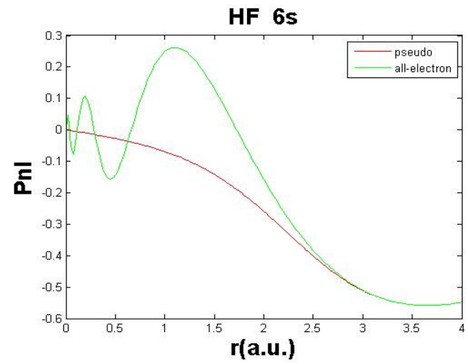


Figure 617.

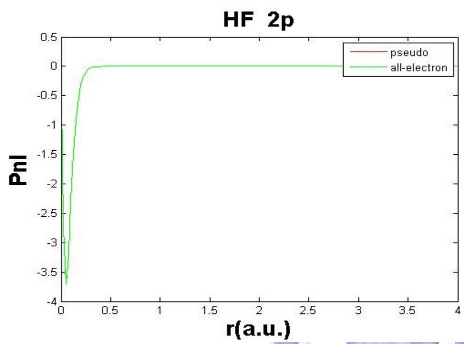


Figure 618.

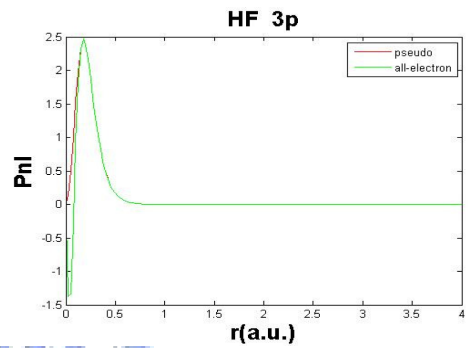


Figure 619.

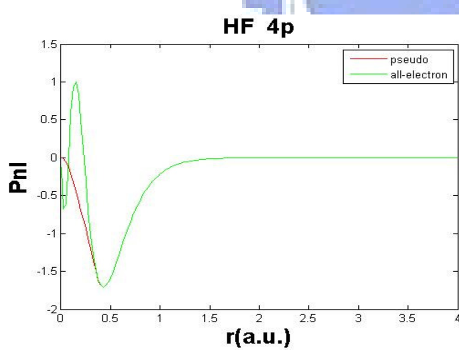


Figure 620.

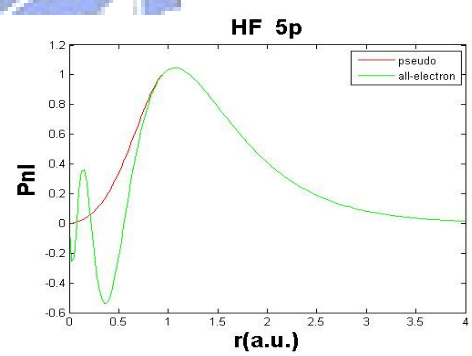


Figure 621.

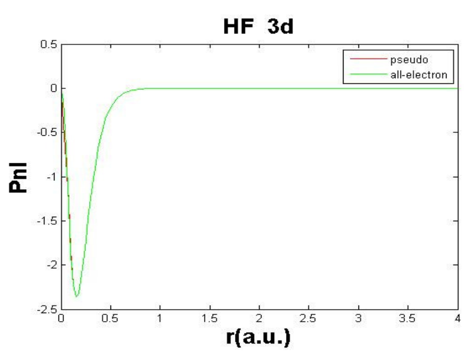


Figure 622.

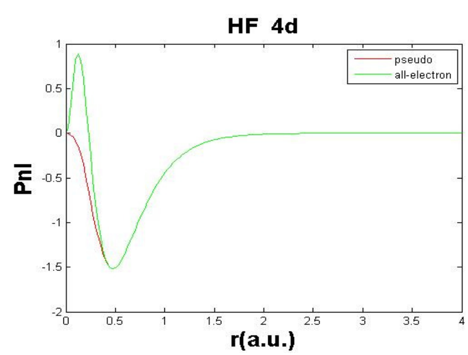


Figure 623.

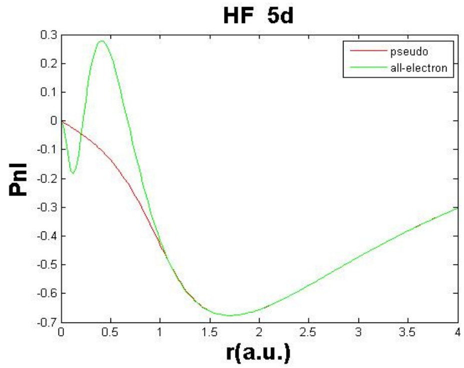


Figure 624.

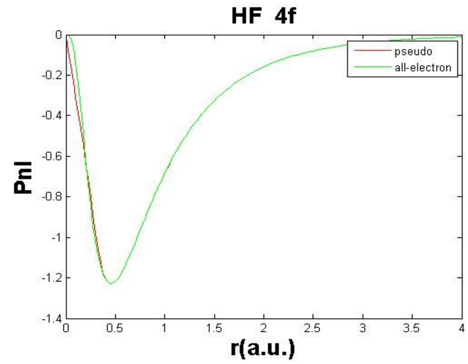


Figure 625.

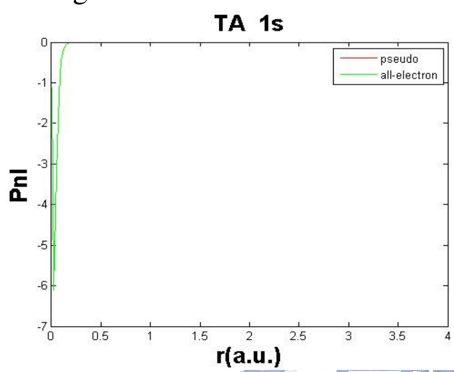


Figure 626.

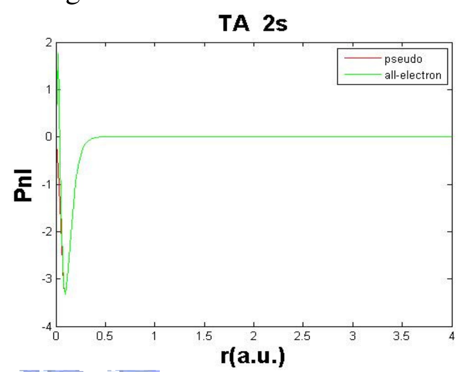


Figure 627.

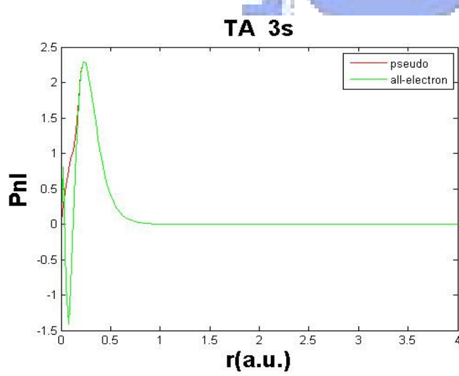


Figure 628.

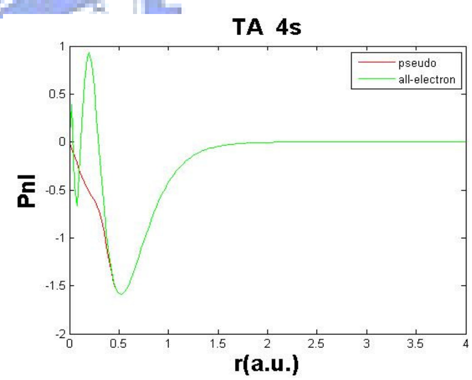


Figure 629.

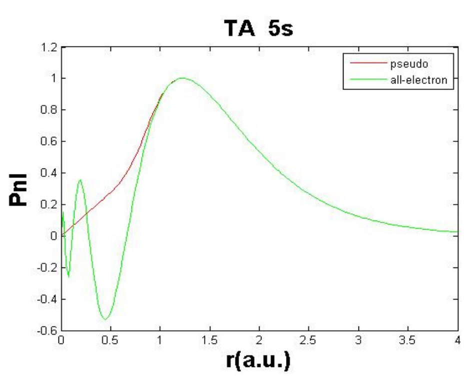


Figure 630.

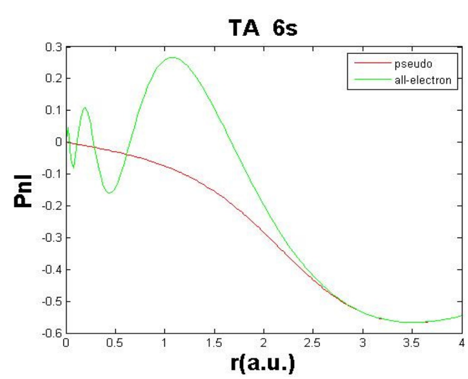


Figure 631.

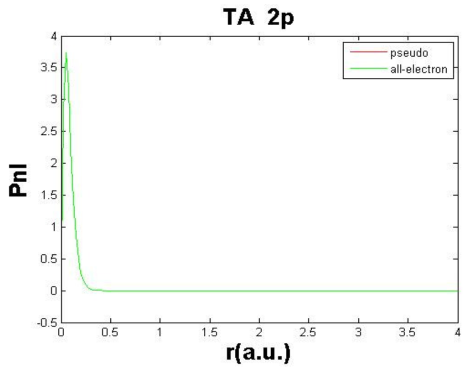


Figure 632.

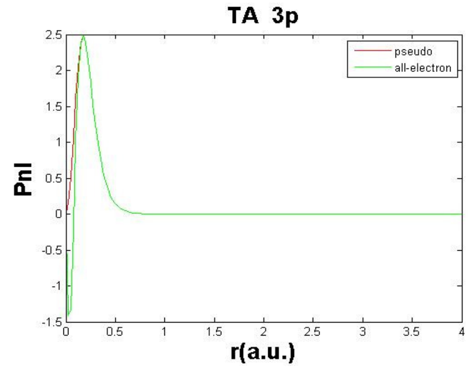


Figure 633.

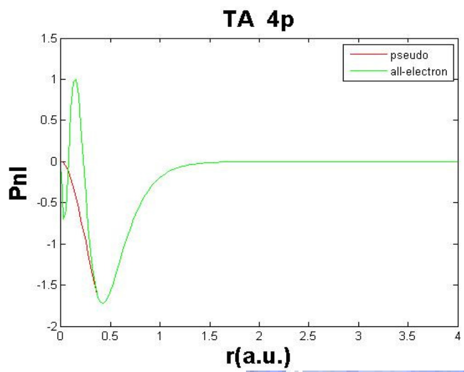


Figure 634.

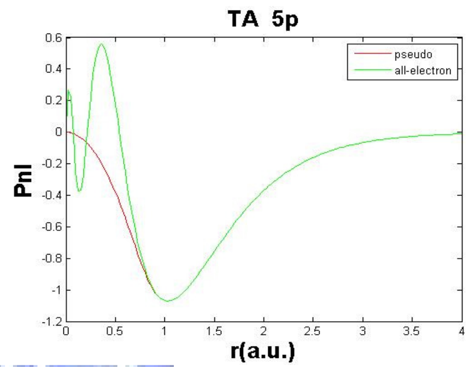


Figure 635.

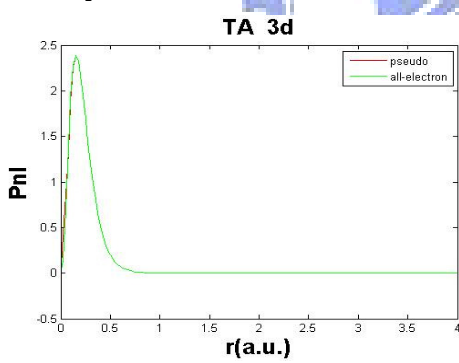


Figure 636.

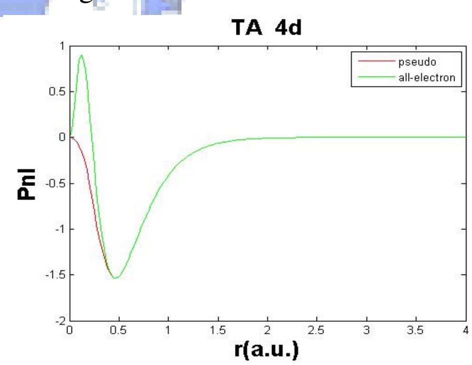


Figure 637.

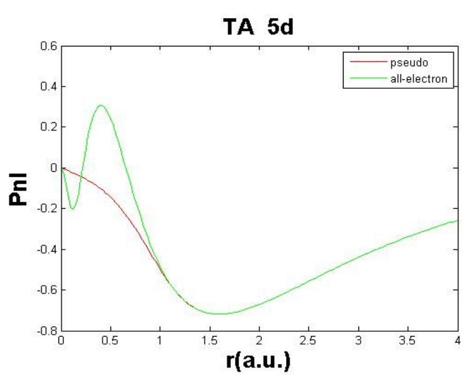


Figure 638.

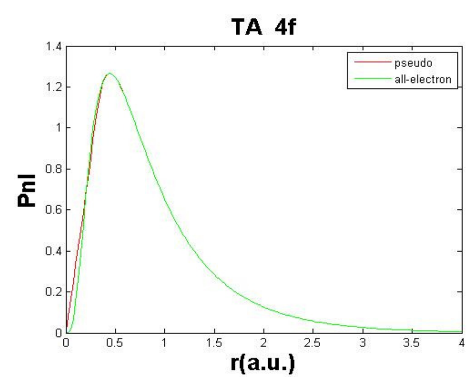


Figure 639.