

Ab initio Computation Revisited: Fweb Based Code for Educational Purposes

Hermawan K. Dipojono
Computational Materials Science Laboratory
Department of Engineering Physics ITB
Jln. Ganesa 10 Bandung 40132
dipojono@tf.itb.ac.id

Abstract

We present an fweb based ab initio computation for educational purposes. As such we emphasized more on simplicity but without loss of generality and on readability than on efficiency of the code for this nontrivial problem. In order to check that our code worked we have tested the code to evaluate the total energies for some elements. We used our code to evaluate the total energies of hydrogen, carbon, oxygen atoms and O^+ . The total energies from our computation were compared to the available data from US National Institute of Standard and Technology (NIST). The results showed that our fweb based ab initio code has performed excellently. For the development of the code we used the density functional theory with local density approximation which gave a relatively simple but accurate prescription for the total energy of a system of electrons and nuclei. The rate of convergence to the final total energy will be presented in terms of total energy versus number of iterations with mixing factor α as a parameter. In the sense of self-consistent computation α is a measure of mixing between the previous and the new density value in order to get the density value for the next step computation in the iteration. Our computation shows that $\alpha = 0.55$ gives the best performance.

Keywords: Ab initio, Simplicity, Readability, Density functional.

1. Introduction

Ab initio computation is a well-known and a widely used for very many contemporary calculations done for solids, such as for atomic structure computations¹⁾, and recently it has been increasingly applied to molecules, including to nanostructure computations²⁾. At the core of this computational approach, once the basic operators have been implemented, the task of developing and debugging is simplified to checking the formulae which have been entered into the software. Although this final stages of the development of the code looks as the most easiest part however this can be the most frustrating one. Looking a bug in hundred of pages code can be months of stressing jobs if it is not done systematically. A unified symbol in the analytical text and the code can contribute significantly in checking the correctness of formulae which have been entered into the code. This in turn will reduce the time spending for debugging the code considerably. From the educational perspectives a unified symbol will provide an easy explanation to the students of how the *ab initio* theory could be transformed into computational codes in a nice way. We have been using the FWEB³⁾ to achieve that objectives. It enables us to maintain both documentation and source code in a single place and is intimately integrated with LATEX. As for the *ab initio* it self we used the density functional theory with local density approximation, one of the standard tools for computation of quantum physical systems.

2. The Total Energy

The total energy of the nuclei and electrons that we want to use is the one developed by Hohenberg, Kohn, and Sham^{4,5)}, i.e. the total energy within density functional theory. The approach we are going to use here is the one suggested by Arias⁶⁾ in his enlightening notes

on the *ab initio* theory of molecules and solids. There are two basic types of energies we have to consider, the potential energy and the kinetic energy of the electrons. We assume the Born-Oppenheimer approximation, assuming the nuclei as stationary. The potential energy can basically be described by classical electrostatic theory. We derived the total potential energy due to the interactions of nuclei with nuclei, of electrons with nuclei, and of electrons with electrons. The total kinetic energy of the electrons is derived as the sum over orbitals of the number of electrons in each orbital times the kinetic energy of each orbital. For the exchange-correlation correction we used the usual practice, that is, the local density approximation. For the exchange-correlation energy per electron, as suggested by NIST, we used the form given by Vosko, Wilk, and Nusair (VWN)⁷⁾.

Knowing all the elements of the total energy we may now write it as follows

$$E[\psi(\mathbf{r})] = \sum_i f_i \int \psi_i^*(\mathbf{r}) \left(-\frac{\hbar^2}{2m} \nabla^2 \psi_i(\mathbf{r}) \right) dV + \int V_{nuc}(\mathbf{r}) \rho(\mathbf{r}) dV + \frac{1}{2} \int \phi(\mathbf{r}) \rho(\mathbf{r}) dV + \int f_{xc}(\rho(\mathbf{r})) dV + U_{nuc-nuc} \quad (1)$$

where $\psi_i(\mathbf{r})$ is Kohn-Sham orbitals which describes the electrons, V_{nuc} is the potential energy of a single electrons at position \mathbf{r} due to the nuclei, $U_{nuc-nuc}$ is the potential energy arising from the interactions among all the nuclei, $\phi(\mathbf{r})$ is the total potential for a single electron at point \mathbf{r} , and $\rho(\mathbf{r})$ is the total electron density given by

$$\rho(\mathbf{r}) = \sum_i f_i |\psi_i(\mathbf{r})|^2 \quad (2)$$

where f_i is the number of electrons in orbital i . The equation (1) is a functional of the orbital functions $\psi_i(\mathbf{r})$, that is, a mapping of each possible choice of the set of electronic orbitals $\{\psi_i(\mathbf{r})\}$ to a unique value for the energy of the system. The problem we want to solve is basically

looking for the correct orbitals: those minimizing the total energy of equation (1) while obeying the quantum orthonormality constraints. It is well known that this optimization may be solved among others by using Lagrange-multiplier equations for constrained minimization. These equations can be derived by taking the derivative of the total energy functional with respect to the Kohn-Sham orbitals. It is easy to show that by using the calculus of variations one may get

$$-\frac{\hbar^2}{2m}\nabla^2\psi_i(\mathbf{r}) + [V_{nuc}(\mathbf{r}) + \phi(\mathbf{r}) + f'_{xc}(\rho(\mathbf{r}))]\psi_i(\mathbf{r}) = \frac{\lambda_i}{f_i}\psi_i(\mathbf{r}) \quad (3)$$

where λ_i is the Lagrange multiplier and $f'_{xc} = \delta E_{xc}/\delta n(\mathbf{r})$, to a first order approximation. The equation (3) is the form of the standard Schrödinger equation for which there are standard techniques to solve it. Note that since the Lagrange-multipliers are unknown constants at the start we may approach in terms of the $\epsilon_i \equiv \lambda_i/f_i$ instead, which is known as the Schrödinger energies for each orbital. The strategy to find the total energy, as suggested by [6], is shown in Figure 1 below. The density must be self-consistent: the density $\rho_{in}(\mathbf{r})$ which we input must lead to a potential $V(\mathbf{r})$ which gives rise to a set of orbitals $\psi(\mathbf{r})$ that sum to a final density $\rho_{out}(\mathbf{r})$ equal to the input density. In our case self-consistency is the condition that $F[\rho(\mathbf{r})] - \rho(\mathbf{r}) = 0$.

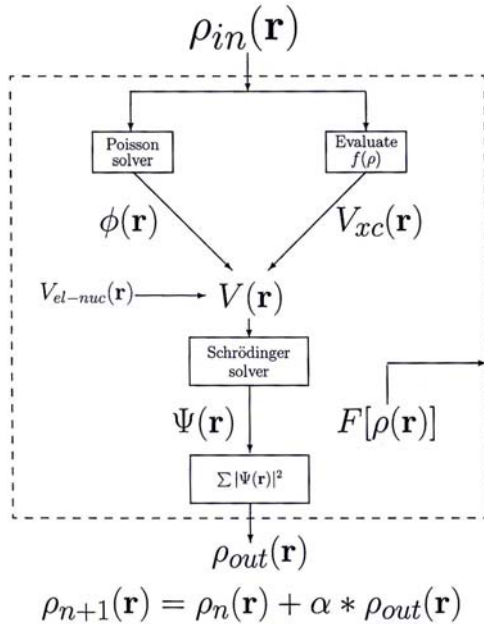


Figure 1. Stages of solving the Kohn-Sham equations.

3. FWEB Based Code

Fweb is a system for *literate programming* developed by Krommes³⁾ based on CWEB system previously devised by Knuth and Levy⁸⁾. It enables one to maintain both documentation and source code in a single place (the web file). FWEB have enabled us to intimately integrate the code with LAT10C. From this in turn, one may gain many advantages such as unified greek symbols

with book-quality typesetting in the code and extensive crossreferencing facilities. Hence increasing the readability of our code. Figure 2 shows a fragment of our poisson code with a unified, an easy to read, and a consistent symbol with the one in the text. The corresponding poisson.web is shown in Figure 3. A source code in C, that is poisson.c, can be generated by ftangle poisson.web. The Figure 2 itself is a postscript file that was generated from poisson.web in Figure 3 by first generating poisson.tex with fweave poisson.web then was processed, as usual, by latex. The only file we have to create is the poisson.web and the rest is basically done by the compiler. Hence reducing the possibility of making error due to typing.

```

3:07

poisson
"poisson.c" 3.3.1 ==

/* derivs_Poisson() for solution of ∇²φ1s(x) = -4πρ(x)
*/
void derivs_Poisson(double x, double φ[], double dφdx[],
int k, double ρ[], double r[], double dr[], int N)
{
double π = 4. * atan(1.);
dφdx[1] = φ[2] * dr[k]; /* Here we use dφdx[1] = dφ/dx */
dφdx[2] = -4.0 * π * ρ[k] * r[k] * dr[k]; /* dφdx[2] = dφ/dr */
}

main( )
{
/* Working variables */
double *y, *φ, *dφdx;
/* Allocate NR vectors for maximum size used */
ρ = dvector(0, Nmx);
φ = dvector(1, 2);
dφdx = dvector(1, 2);
free_dvector(ρ, 0, Nmx);
free_dvector(φ, 1, 2);
free_dvector(dφdx, 1, 2);
}

```

Figure 2. A poisson.ps file generated from poisson.web in Figure 3.

One can easily see from Figure 2 of how the greek symbols are incorporated into the program. For example, the common practice being done in the programming is that if one wants to write ψ_i one has to write psi_i instead. There is nothing wrong here, of course, but if we have so many formulae in greek alphabets, the one like in *ab initio* molecular dynamics simulations, debugging and checking the formulae which have been entered into the program of hundred pages would be quite difficult. The proximity between program fragments and their explanations (the analytical in the text) would help to reduce such difficulty. The ultimate goal of using FWEB is the detachment of program explanation structure from program structure. In other words reading the program is as easy as the analytical text.

```

@c/ @% / ***** /
@* poisson.
@a

/* derivs\_Poisson() for solution of  $\nabla^2\phi_{is}$ 
(\mathbf{x}) = -4\pi \rho(\mathbf{x})$ */
void derivs_Poisson(double x, double phi[], double dydx[],
int k, double rho[], double r[], double dr[], int N)
{
    double pi=4.*atan(1.);
    /* Here we use  $\frac{d\phi}{du}[1] = \frac{d\phi}{du}$  */
    dydx[1]=phi[2]*dr[k];
    /*  $\frac{d\phi}{du}[2] = \frac{d\phi'}{du}$  */
    dydx[2]=-4.0*pi*rho[k]*r[k]*dr[k];
}

main()
{
    /* Working variables */
    double *y, *phi, *dydx;
    /* Allocate NR vectors for maximum size used */
    rho=dvector(0,Nmx);
    phi=dvector(1,2);
    dydx=dvector(1,2);
    free_dvector(rho,0,Nmx);
    free_dvector(phi,1,2);
    free_dvector(dydx,1,2);
}

@*1 .
@W rho "\rho"
@W pi "\pi"
@W phi "\phi"
@W Phi "\Phi"
@W psi "\psi"
@W Psi "\Psi"
@W dydx "\frac{d\phi}{du}"
@* \INDEX.

```

Figure 3. A poisson.web file of the postscript file in Figure 2.

4. Computational Results

To show that our code works we use it to calculate the total energy of hydrogen, carbon, oxygen atoms and O^+ . Figure 4 shows the convergence of our computational results to the expected one given by NIST. The total energy is evaluated by varying the α , a measure of mixing between the previous and the next density as shown in Figure 1. We found that $\alpha = 0.55$ gave the best performance and could not increase the a further without increasing the number of iterations to reach the correct value. Table 1 shows the value of the components of the total energy from our computation and the one from NIST. It shows that our computational results are in excellent agreement with the corresponding results from NIST.

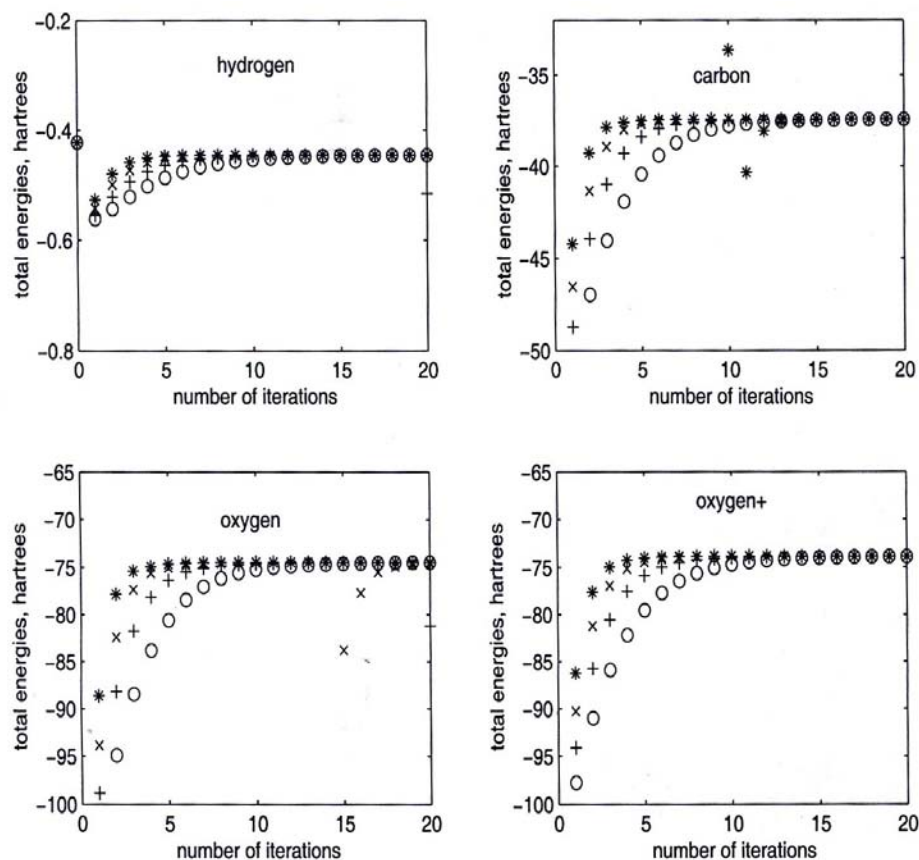


Figure 4. Number of iterations vs total energies, in hartrees, with α as a parameter for some elements:
 $\circ \rightarrow \alpha = 0.25$, $+$ $\rightarrow \alpha = 0.35$, $\times \rightarrow \alpha = 0.45$, and $*$ $\rightarrow \alpha = 0.55$.

Table 1. Energies, in hartrees, for some elements for $\alpha = 0.55$, iteration number 20

Energies	hydrogen		carbon		oxygen		oxygen+	
	NIST	Ours	NIST	Ours	NIST	Ours	NIST	Ours
E_{total}	-0.445671	-0.4456705	-37.425749	-37.426204	-74.473077	-74.473077	-73.863138	-73.863139
$E_{exchange}$	-0.232525	-0.232525	-4.728724	-4.728733	-7.768482	-7.768482	-7.320043	-7.320043
$E_{kinetic}$	0.425027	0.425027	37.190391	37.190424	74.116881	74.116881	73.538518	73.538518
E_{Coul}	0.282827	0.2828268	17.627997	17.627603	36.331102	36.331101	31.687656	31.687655
E_{elnuc}	-0.920999	-0.920999	-87.515412	-87.515499	-177.152578	-177.152578	-171.769269	-171.769269

5. Conclusions

We have developed a fweb based code for *ab initio* computation and have used the density functional theory with local density approximation to calculate the total energy of the system of nuclei and electrons. FWEB code has enabled us to incorporate the greek symbols into the code and hence increasing the readability of our code. To show that our code worked we have used it for evaluating the total energies of hydrogen, carbon, oxygen atoms and O^+ . By using the exchange-correlation energy form of VWN we found that the total energies for hydrogen, oxygen, carbon and O^+ were in excellent agreement with those from NIST. These results were achieved with the mixing parameter $\alpha = 0.55$.

Acknowledgement

This work is partially supported by the Department of National Education of the Republic of Indonesia under Hibah Bersaing XI. The author would also like to acknowledge Prof. Tomas Arias of Cornell for his enlightening notes on *ab initio* computation for molecules and solids.

References

1. Q. M. Zhang, G. Chiarotti, A. Selloni, R. Car and M. Parrinello, "Atomic structure and bonding in liquid GaAs from *ab initio* molecular dynamics", *Phys. Rev. B*, **42.8**, 5071, (1990).
2. Antonis Andriotis, Madhu Menon, and George Froudakis, "Catalytic action of Ni atoms in the formation of carbon nanotubes: A combined ab-initio and molecular dynamics study" in *Microelectronics, Microsystems and Nanotechnology*, papers presented at MMN 2000, ed. Androula G. Nassiopoulou and Xanthi Zianni, World Scientific, 2001
3. John A. Krommes, 'FWEB: A Web system of structured documentation for multiple languages', <http://w3.pppl.gov/krommes/fweb.html> Oct 1 2003.
4. P. Hohenberg and W.Kohn, "Inhomogeneous Electron Gas", *Phys. Rev.*, **136.3B**, B864 (1964).
5. W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects", *Phys. Rev. B*, **140. 4A**, A1133, (1965).
6. Tomas A. Arias, Notes on the *ab initio* theory of molecules and solids: Density functional theory (DFT), Department of Physics, Cornell University, 2000.
7. S.J. Vosko, L. Wilk, and M. Nusair, "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis", *Can. J. Phys.* **58**, 1200-1211, (1980).
8. Donald E. Knuth and Silvio Levy, *The CWEB System of Structured Documentation*, Addison-Wesley, 1993.