Theoretical Material Science: Electronic structure theory at the computer

Prepared by Björn Bieniek and Volker Blum
Based on an exercise by Viktor Atalla, Oliver Hofmann, Sergey Levchenko
Berlin, April 2012

Some rules on expected documentation from this exercise.

The computational exercises are intended as "hands-on" experience with actual, numerical electronic structure theory. Our main goal is to fill some of the basic concepts with life for real systems.

If you are fast, you may finish the exercise below in the actual exercise class. If not, we ask you to try finish as much as you can, and then come back at another point, follow the script, and finish the exercise.

You can do the actual computations in pairs of two, if you wish. However, please still hand in your solutions to the exercise individually, as you would normally do. What is expected is a record of the basic data that we ask for (e.g., in table form), rough answers to the questions asked (answers can be short, but should be there, and should indicate that you understood the meaning of your data), and plots, where required.

For required plots, please use the printer in the PC pool and append them to your exercise materials. As you may know, printed pages usually cost 5 cents per page on that printer. *However*, for the purposes of the exercise, a contingent of free pages has been agreed upon. If you are doing printouts for this class, please mention this to the administrators.

Please note that the page contingent is not gigantic – so be careful and do not print excessively many pages. A few pages at most should be sufficient for the solutions, anyway.

Please hand in all your solutions as usual at the beginning of the exercise in the following week.

Contents

1 Background

With Hartree-Fock theory and methods based on it, as well as with density-functional theory (DFT), we have theoretical methods that provide much insight into the many-particle problem. In addition, they allow us to make actual predictions for real materials at the computer. This achievement is no small feat—electronic structure theory is "big business" today in academic and industrial materials science. The numbers of citations to the papers describing the most successful basic approximations testify to their popularity (in several cases, well above 10,000 or even 30,000).

While pure Hartree-Fock theory is mostly used as a starting point for other methods today, DFT is used in practice in many flavors. In the present exercise, we focus only on the simplest one, the local-density approximation (LDA). The basic framework is given by the Hartree-Fock- and Kohn-Sham-equations:

HF-equation:

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + v_{ext}\left(\mathbf{r}\right) + v_H\left(\mathbf{r}\right)\right]\varphi_m\left(\mathbf{r}\right) - 2\frac{e^2}{4\pi\epsilon_0}\sum_n^N \int \mathrm{d}\mathbf{r}' \frac{\varphi_n^*\left(\mathbf{r}'\right)\varphi_m^*\left(\mathbf{r}'\right)\varphi_n\left(\mathbf{r}\right)}{\left|\mathbf{r}-\mathbf{r}'\right|}\varphi_m\left(\mathbf{r}\right) = \epsilon_m\varphi_m\left(\mathbf{r}\right) \tag{1}$$

KS-equation:

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + v_{ext}(\mathbf{r}) + v_H(\mathbf{r}) + v_{xc}(\mathbf{r})\right]\varphi_m(\mathbf{r}) = \epsilon_m\varphi_m(\mathbf{r}) \tag{2}$$

We could attempt to solve these equations on our own, e.g., by writing a computer code. For a general-purpose solution (different types of atoms in different places, non-periodic or periodic boundary conditions, convergence of all numerical approximations), however, this would be a lot of work. The code we will be using today is one of many possible general-purpose electronic structure codes available today, and happens to be developed in Berlin: the "Fritz Haber Institute *ab initio molecular simulations*" (FHI-aims) package. As a whole, this code is significantly more powerful than what is needed in the exercise today. Yet, it is a "living" piece of code. Although it is big, adjustments to the source code for specific tasks are frequent, and ongoing. You are welcome to look at the source or even modify it if you are interested. However, please do not pass it on.

2 A quick tour of the pieces needed in an electronic structure code

Solving the original real-space equations above by brute force on a grid is, however, not practical either. Instead, most electronic structure codes today make specific numerical choices that are also reflected in the input files which we will edit further below. These choises are:

Basis set:

Instead of writing the orbitals φ_m directly on a three-dimensional grid, it is advantageous to use an expansion in terms of a quantum-mechanical *basis set* $\{\chi(\mathbf{r})\}$—a set of functions of a given shape that is "complete enough" to represent the orbitals, density etc, accurately. We expand:

$$\varphi_m(\mathbf{r}) = \sum_j c_{mj}\chi_j(\mathbf{r}) \tag{3}$$

There are many possible choices for the shape of the set $\{\chi(\mathbf{r})\}$: Gaussian-type functions, so-called "Slater-type orbitals", plane waves, more complex (and more accurate) combined objects known as "augmented plane waves", even grid-based functions. In FHI-aims, we use atom-centered functions of the following general shape:

$$\chi(\mathbf{r}) = \frac{u(r)}{r}Y_{l,m}\left(\theta,\phi\right) \tag{4}$$

For each basis function, we thus need to specify l (all allowed m are included implicitly), and the shape of the radial function $u(r)$, which could be "any shape" ($u(r)$ are numerically tabulated functions). For the first exercise, we will not be concerned so much with the detailed choice of

this shape. Adequate sets of basis functions are provided, and our primary task will be to let the FHI-aims know (via an input file) which of these sets we would like to use.

Nonetheless: Next to the exchange-correlation treatment itself, the choice of a basis set can constitute the primary approximation in a practical calculation. For sure, the exact choice of basis functions is very closely related to the accuracy and efficiency that can be reached in a given code.

Numerical integration:

For a general single-particle equation

$$\hat{h}\varphi_m(\mathbf{r}) = \epsilon_m \varphi_m(\mathbf{r}) \quad , \tag{5}$$

we can insert the basis-set expansion of $\varphi_m(\mathbf{r})$ above, and then additionally integrate both sides from the left with $\int d^3\mathbf{r}\chi_i(\mathbf{r}\cdot[...]$. We obtain:

$$\sum_j h_{ij} c_{mj} = \sum_j \epsilon_m s_{ij} c_{mj} \quad , \tag{6}$$

where

$$h_{ij} = \langle \chi_i | \hat{h} | \chi_j \rangle \tag{7}$$

$$s_{ij} = \langle \chi_i | \chi_j \rangle \, . \tag{8}$$

h_{ij} and s_{ij} are known as the Hamiltonian and overlap *matrices*, respectively. Our goal is still to find $\varphi_m(\mathbf{r})$ and ϵ_m, but the problem is now reduced to one of finite-dimensional linear algebra. In practice, it is obviously necessary to perform the numerical integrals $\int d^3\mathbf{r}[...]$ for h_{ij} and s_{ij}, but the details will not concern us today. Just do not be surprised if you encounter specifications for numerical integration grids somewhere in the input file.

Generalized eigenvalue problem:

Equation (6) is known as a generalized eigenvalue problem and can be solved by standard linear algebra (e.g., the Lapack library or parallel versions thereof).

Hartree potential:

In all cases, $\hat{h}$ contains the so-called Hartree of electrostatic potential of the density n. Finding the Hartree potential is an exercise in solving Poisson's equation, which we shall also not revisit here. We only mention that FHI-aims uses a decomposition of the Hartree potential in atom-centered multipole components, very much like what is taught in classical electrodynamics. Do not be surprised if you encounter the maximum expansion order l_{hartree}, in the input files.

Self-consistency:

This is perhaps the biggest technical part of any computation: We have all the pieces to compute $\{\varphi_m\}$ and $\hat{h}$, but $\hat{h}$ depends on $\{\varphi_m\}$, and $\{\varphi_m\}$ depend on $\hat{h}$.

With some mathematical training, we recognize that this setting actually defines a non-linear search problem, with stationary points where a given $\hat{h}$ defines a solution $\{\varphi_m\}$ which, in turn, yields $\hat{h}$. We need to find such a stationary point ("self-consistent solution"). In general, there could be more than one, or even a divergent solution, depending on where we start our search. In practice, one successful strategy is to iterate to convergence:

1. Start with an initial guess for $\{\varphi_i\}$ (we could just as well start with $n(\mathbf{r})$

2. Calculate $n\,(\mathbf{r})$, perhaps $n\,(\mathbf{r}, \mathbf{r}')$, and the Hartree-Fock or Kohn-Sham potentials and their matrix elements

3. Solve the HF / KS equations to obtain a new set of $\{\varphi_m\}$

4. Check for convergence. Normally, we compare the density from one iteration to the next, the sum of single-particle eigenvalues from one iteration to the next, and the total energy from one iteration to the next. If all three change only below a given set of tolerances, the calculation is considered converged. If one or more of them change more than a given tolerance, we go back to the second step and repeat the cycle.

3 The first exercise: Setup, in steps (8a)

To get comfortable with the code, we will perform the first exercise in simple steps, explaining what is needed as we go along. The text of this exercise (8a on the exercise sheet) is:

> 8a Using a so-called "minimal basis" (only one basis function, valid for hydrogen), perform a self-consistent Hartree-Fock calculation for the Hydrogen atom. What are the single-particle energies and total energies? How close are they to the result that you expect? What could be the origin of the difference?

In other words, we need to learn how to tell the code:

- that we are dealing with a Hydrogen atom
- what a "minimal basis" is (what is that, anyway?? We will see in a moment ...)
- that we are interested in results for the Hartree-Fock method

and how to run the code in practice, and how to read the output that the code produces.

If you are interested to learn more, there is also a manual "/media/public/TFKP_2012/FHI-aims.pdf" to be found on the computers in the PC pool. However, you should not need the manual for this exercise. All the details relevant here are explained below.

The basic protocol is simple:

- **Run each calculation in its own directory**
- **There are only two input files, to be edited as ASCII text**
  control.in
  geometry.in
- **Run the calculation in its directory, at the command line**
  `mpirun -np 1 aims.VERSION.mpi.x | tee calculation.out`

Do not try out the last line just yet. Note that the "mpirun" command allows you to parallelize the code, normally over more than one processor. We here use it only for a single processor ('-np 1'), but we may use two processors for later exercises.

In detail, some preparation of the computer is needed:

You will have to do these steps only once, but we will need to set some required defaults for the operating system.

At the computer, open a terminal window.

In the /media/public/TFKP_2012directory, there is a file called "bashrc_sample". Copy this file to your home directory as a file called ".bashrc", OR incorporate the relevant lines into an existing ".bashrc" file.

If you are not running the "bash" shell by default (this depends on your PC pool account), just perform the exercise after typing "/usr/bin/bash" at the command line.

First, create a working directory and change into that directory (using the 'mkdir' and 'cd' commands). The 'ls -l' command will produce a file listing of that directory (hopefully, it is empty).

We next create the input files, using a text editor ('emacs', 'vi', an editor from the menu, or whatever you are comfortable with).

3.1 geometry.in

The `geometry.in` file contains all information concerning the atomic structure of the system. This includes the nuclear coordinates, which are specified by the keyword `atom`, followed by cartesian coordinates (in units of Å) and the descriptor of the species. For example, the input

```
atom 0.0 0.0 0.0 H
```

specifies a single hydrogen atom at the origin.

3.2 control.in

This file contains all other settings for the calculation. In particular, it specifies the physical and technical settings for the equations to be solved. A minimal example, which can be used as a template during the exercise, would be

```
xc          hf
charge      0
spin        collinear
default_initial_moment hund
```

```
sc_accuracy_eev   1E-2
sc_accuracy_rho   1E-4
sc_accuracy_etot  1E-5
sc_iter_limit     300
```

- `xc`
  This keyword sets the method to be used. Here, it is set to `hf`, which requests a Hartree-Fock calculation.
- `charge`
  Set the total charge of the system in units of $|e|$. For a neutral system, this is zero.
- `spin`
  This keyword governs the spin treatment. It can be set to `none`, which requests a spin-restricted (unpolarized) calculation, or to `collinear`, which requests a spin-unrestricted (polarized) calculation.
- `default_initial_moment hund`
  Sets the initial spin of the atoms. The value "hund" requests an initial moment taken from the aufbau principle. Only necessary for `spin collinear` calculations.

The other lines set the convergence criteria of the self-consistent cycle regarding the change of the density (`sc_accuracy_rho`), the sum of eigenvalues (`sc_accuracy_eev`), and the total energy (`sc_accuracy_etot`) between two consecutive cycles. For the energies, the units are 'eV'.

The remaining block of `control.in` defines all computational parameters associated with the elements (species) – most importantly the basis set. Each element listed in the `geometry.in` must also be listed in `control.in`. The order is not important.

FHI-aims provides pre-defined settings as a starting point for all species. Defaults are provided for three different levels of accuracy, light, tight, and really tight.

These settings actually specify a number of technical settings for each element that we want users to know about. Therefore, we do not hide them in the code itself, but rather require them to be

part of the input file `control.in`. Such settings include integration grids, the expansion order of the Hartree potential, or the actual basis set. For the purposes of this exercise, as well as for many other purposes, you will only have to manipulate a small part of these settings. Thus, the standard practice is to predefine working settings that you can simply copy into `control.in`. They work, and you will only have to change what is specifically used in this exercise.

These "species default" files can be found in the directory /media/public/TFKP_2012/species_defaults and should be copied and pasted into `control.in`, e.g. via the command

```
> cat /media/public/TFKP_2012/species_defaults/tight/01_H_defaults >> control.in
```

The basis functions associated with a given species are also tabulated at the end of these default settings:

```
# "First tier" - improvements:  -1014.90 meV to -62.69 meV
    hydro 2 s 2.1
    hydro 2 p 3.5
# "Second tier" - improvements:  -12.89 meV to -1.83 meV
    hydro 1 s 0.85
    hydro 2 p 3.7
    hydro 2 s 1.2
    hydro 3 d 7
# "Third tier" - improvements:  -0.25 meV to -0.12 meV
#   hydro 4 f 11.2
#   hydro 3 p 4.8
#   hydro 4 d 9
#   hydro 3 s 3.2
```

Strictly speaking, each line above defines a single radial function $u(r)$ with given main quantum number n and angular momentum quantum number l. The code then automatically adds the full set of possible basis functions $u(r)/rY_{lm}(\theta,\phi)$, i.e., the functions for all $m{=}-l,\ldots,+l$. In the case given above, all listed basis functions are derived from Hydrogen-like central potentials $-Z/r$, where the last number given corresponds to the effective nuclear charge Z used in the generation of $u(r)$.

Lines that begin with a '#' symbol are commented out and will be ignored. Additional basis functions can be included in the calculation by uncommenting the corresponding lines. The basis functions are classified in "tiers" (which means "levels"). Systematically improved calculations can be performed by enabling additional tiers.

Note that, in addition to the list above, FHI-aims *also* includes the radial function of the non-spinpolarized, spherically symmetric free atom itself, and based on density functional theory, not Hartree-Fock. We will be using these functions as the "minimal basis" set below.

The basis functions of the free atom are taken only for the occupied levels. They are therefore also called the "minimal basis", as the eigenstates of the free atom can not be correctly represented with fewer basis functions than the minimal basis. In the case of Hydrogen, there is only one radial function of $1s$ character in the minimal basis.

A calculation based only on the minimal basis can be set up by commenting out all *other* basis functions. In the example above, this would mean that all the uncommented lines in the "first tier" and "second tier" need to be commented out as well.

3.3 Now really: The first exercise (8a)

Exercise 1 (**The hydrogen atom**)**.** We first focus on the simplest atom, where the exact solution (of Schrödinger's Equation) is known.

1. Generate a simple `geometry.in` file by hand, which contains only a single hydrogen atom.

2. Generate a simple `control.in` file by hand. Request a calculation of an uncharged, spin-polarized hydrogen atom using Hartree-Fock as the exchange-correlation method (see boxes above).
   Additionally, don't forget to set convergence criteria for the SCF cycle, also given above.
   Finally, paste the "tight" species data of H into the `control.in` file, e.g. via the command
   `> cat /media/public/TFKP_2012/species_defaults/tight/01_H_default >> control.in`

Again, as we are aiming for a *minimal basis* calculation, comment out the basis functions of the "first tier" and "second tier" as well.

Now, run FHI-aims:
`> mpirun -np 1 aims.VERSION.mpi.x | tee H.out`
The "tee" command is a command that will have your output printed on the screen before you, in addition to saving it in a file called 'H.out' in the example above.

When done, open the output file. If you find the line "Have a nice day" at the end, then your calculation is converged.

The output file of FHI-aims contains a lot of information. Basically, it is a chronicle of the preparatory steps, the initialization of the self-consistency cycle, and of each iteration of the self-consistency cycle up to convergence.

We are interested in the total energy and eigenvalues at convergence. The final values are printed during the last self-consistency iteration. Thus, in the output file, find the last blocks of this form:

```
Total energy components:
| Sum of eigenvalues            :          -0.49602402 Ha         -13.49750036 eV
| XC energy correction           :           0.28306053 Ha           7.70246903 eV
| XC potential correction        :           0.00000000 Ha           0.00000000 eV
| Free-atom electrostatic energy:          -0.28276797 Ha          -7.69450788 eV
| Hartree energy correction      :          -0.00029255 Ha          -0.00796078 eV
| Entropy correction             :           0.00000000 Ha           0.00000000 eV
| ---------------------------
| Total energy                   :          -0.49602401 Ha         -13.49749998 eV
| Total energy, T -> 0           :          -0.49602401 Ha         -13.49749998 eV
| Electronic free energy         :          -0.49602401 Ha         -13.49749998 eV

Writing Kohn-Sham eigenvalues.

Spin-up eigenvalues:

State    Occupation    Eigenvalue [Ha]    Eigenvalue [eV]
    1       1.00000          -0.496024          -13.49750

Spin-down eigenvalues:

State    Occupation    Eigenvalue [Ha]    Eigenvalue [eV]
    1       0.00000           0.070097            1.90744
```

Again, look at the total energy (the line labelled simply "Total energy"!) and eigenvalues. How close are they to the result that you expect? What could be the origin of the difference?

4 The second exercise (8b)

You are now an expert in first-principles calculations with FHI-aims. The next step is to vary the exchange-correlation treatment, leaving all other settings constant.

We are aiming for the local density approximation (LDA) to Kohn-Sham DFT. While this is (in principle) a unique functional, in practice there exist several different parametrizations. In your control.in file, change the following line to read:

```
xc          pw-lda
```

This change will invoke the local density approximation to the exchange functional as first given by Slater, and additionally the correlation energy of the electron gas in a parametrization published by Perdew and Wang in 1991.

After running the calculation, evidently the resulting values are different. Did you expect this change? Can you rationalize it?

5 The third exercise (8c)

In this exercise, we basically repeat the calculations for Hartree-Fock and for the LDA. This time, however, we vary the number of basis functions in order to check for basis set convergence. Use the following sets of basis functions:

- The "minimal" basis
- Add the basis functions of the "first tier"
- Add the basis functions of the "second tier"
- Add the basis functions of the "third tier"

Do you observe "convergence"? If so, what converges? Is the variational principle fulfilled? Between LDA and Hartree-Fock, can you identify the "classic" effects of self-interaction?

In general, LDA should not be very good for the H atom. This is, of course, not unexpected (it's the worst case for LDA). In solid state physics, the LDA had a much better track record from the outset, in fact being the first viable method to treat metals with reasonable accuracy. In quantum chemistry, though, the "failures" of LDA especially (but not only) for poster-child systems like the H atom were viewed with much skepticism, and the true breakthrough of DFT in chemistry did not happen until improved functionals became available.

6 The fourth exercise (8d)

In this exercise, we will examine the total electron density of the Hydrogen atom. We do this using viewing program for three-dimensional arrays (n is a function of x, y, and z) and a *de facto* standard format to output such quantities in molecular science, the "cube" format.

Only perform this and the following exercises for a single basis set, including all basis functions up to "tier 2".

Technical note: On the computers at the PC pool, we have observed that the cube file output can sometimes "hang" for unknown reasons. A fix is simple: Interrupting the calculation and rerunning the exact same calculation right away usually works. We suspect that the deeper reason is not within the code, but rather to do with a file system or timeout in the operating system. Apologies for the inconvenience.

To obtain the electron density in the "cube" format, add

```
output cube total_density
cube filename total_density_uncharged.cube
```

to the `control.in` file. By default, FHI-aims will figure out the region to plot into the cube files by itself.

Run the calculation and look at the output. Visualize the density with the *jmol* molecular visualization program.

In jmol, you will have to open the "Console" from the "File" menu. Once you have this console, you can use typed commands to specify which isosurface of the density you would like to see.

Once you have opened "jmol" and the "Console", type:

```
load "total_density_uncharged.cube"
```

Now we still have to tell jmol which "isosurface" of the electron density we would like to see. For example:

```
isosurface cutoff 0.1 "total_density_uncharged.cube"
```

You can play with the cutoff criterion to see how far the density extends away from the atom. To undo a given isosurface before creating the next one, type

```
isosurface delete
```

"jmol" is a much more powerful utility than this, but for the moment, that's all you'll need to know.

Do the calculations with Hartree-Fock and LDA.

Only if you want to play – as an optional check – you are also free to choose the cube region yourself by adding the following lines to `control.in` after the `output cube` keyword:
The line
`cube origin` ***`x y z`***
allows to determine the center of the space to be plotted, with x y z being its cartesian coordinates in Å. The edges of the cube file can be configured using the syntax
`cube edge` ***`n dx dy dz`***,
where n indicates the number of steps of a particular edge (voxel), and dx, dy, dz indicate the length of each individual step in x, y, and z direction, respectively. Note that each cube file, being a 3D-object, requires 3 edges to be completely defined. E.g., a complete definition of a cube file could look like

```
output cube total_density
cube filename total_density.cube
cube origin 0 0 0
cube edge 100 0.1 0.0 0.0
cube edge 150 0.0 0.1 0.0
cube edge 50 0.0 0.0 0.1
```

7 The fifth exercise (8e)

For a spherical atom, we have an even simpler way to visualize where the majority of the electron density is: A simple one-dimensional plot of the integrated density at each distance from the nucleus, $4\pi r^2 n(r)$. The steps are as follows:

We manipulate the output format of the cube file to only write $n(r)$ for a line points along the x axis, yet with an extremely dense grid. We can accomplish this task by adding the following lines to the control.in file of the previous task:

```
output cube total_density
cube filename total_density_x.cube
cube origin 0 0 0
cube edge 9999 0.001050835442 0.0 0.0
cube edge 1 0.0 0.1 0.0
cube edge 1 0.0 0.0 0.1
```

If you are wondering about the strange number above: These are 0.002 Bohr radii (atomic units), written down in Å units. The cube file format (which we did not invent) uses atomic units internally, and this choice will prevent roundoff errors in the actual cube file output.

Based on the resulting .cube file, we still need to multiply the density by $4\pi r^2$. For the setup above, we provide a simple script that accomplishes this task and produces a two-column file with ASCII data. All you need to do is to run the script 'plot_density.pl' as follows:

```
plot_density.pl   total_density_x.cube > density_x.dat
```

Just use any plotting tool (e.g., xmgrace) to plot the ASCII output file 'density_x.dat'. The output units for the radial coordinate are Angstrom. With this knowledge (the unit of the axis), does the plotted density match your physical intuition – and why?

8 The sixth exercise (8f)

We now repeat the same steps as before, but for the Si atom. Actually, we will simplify the steps somewhat: Please only perform all calculations *only* using all basis functions up to and including the "second tier".

In detail, this means:

- Create a geometry.in file and a control.in file for the Si atom, and copying the "tight" Si species defaults into control.in. **Do not forget to enable all basis functions up to the "second tier"**.
- Perform a self-consistent calculation for the Si atom, using the Hartree-Fock method.
- Perform another self-consistent calculation using the 'pw-lda' method.
- Find the converged eigenvectors and total energies from both calculations. Can you spot the valence and core eigenvalues? How different are they for the different treatments?
- Plot the three-dimensional electron densities as before. Do they look different?
- Plot the one-dimensional electron density $4\pi r^2 n(r)$ for both methods, using the same technique as before. What does it look like? Can you interpret the result?