



Questions? — Hudson Hall 235 or Hudson Hall 1111 —

# Free Atoms in DFT

## Hands-On Instructions for the FHI-aims Code

### Contents

<b>1</b>	<b>Theoretical background</b>	<b>1</b>
<b>2</b>	<b>A quick tour of the technical details</b>	<b>2</b>
2.1	Basis set . . . . .	2
2.2	Numerical integration . . . . .	3
2.3	Generalized eigenvalue problem . . . . .	3
2.4	Electrostatic (Hartree) potential of the electrons . . . . .	3
2.5	Self-consistency . . . . .	4
<b>3</b>	<b>Single atom calculation with FHI-aims — Step by step</b>	<b>4</b>
3.1	Task preparation . . . . .	5
3.2	geometry.in . . . . .	5
3.3	control.in . . . . .	5
3.4	The basis set . . . . .	6
<b>4</b>	<b>Finally: The Hydrogen Atom</b>	<b>7</b>
4.1	Energy and Eigenvalues for the Minimal Basis in Hartree-Fock Theory . . . . .	7
4.2	Changing to the Local-Density Approximation . . . . .	8
4.3	Converging the basis set . . . . .	8
4.4	Plotting the density . . . . .	9
4.5	Visualizing the radial distribution function . . . . .	10
<b>5</b>	<b>The Silicon Atom</b>	<b>10</b>

### 1 Theoretical background

*This section summarizes some of the basics that were already taught in class. We hope that this is helpful. The actual exercises, found in later sections, are hopefully simple tasks to carry out.*

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. While pure Hartree-Fock theory is mostly used as a starting point in *quantum chemistry* (high-accuracy molecular calculations), computational materials science today mostly relies on DFT in many flavors.

The basic framework is given by the Hartree-Fock equations (for the non-spinpolarized case, so-called restricted Hartree-Fock)

$$\left[ -\frac{\hbar^2}{2m}\nabla^2 + v_{\text{ext}}(\mathbf{r}) + v_{\text{es}}^e(\mathbf{r}) \right] \varphi_m(\mathbf{r}) - e^2 \sum_n^{N/2} \int d\mathbf{r}' \frac{\varphi_n^*(\mathbf{r}')\varphi_n(\mathbf{r})}{|\mathbf{r} - \mathbf{r}'|} \varphi_m(\mathbf{r}') = \epsilon_m \varphi_m(\mathbf{r}) \quad (1)$$

or the Kohn-Sham equations

$$\left[ -\frac{\hbar^2}{2m}\nabla^2 + v_{\text{ext}}(\mathbf{r}) + v_{\text{es}}^e(\mathbf{r}) + v_{\text{xc}}(\mathbf{r}) \right] \varphi_m(\mathbf{r}) = \epsilon_m \varphi_m(\mathbf{r}). \quad (2)$$

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 are going to use here is one of many possible general-purpose electronic structure codes available today, and happens to be co-developed in our group here at Duke: the FHI-aims package. As a whole, this code is significantly more powerful than what is needed in the exercise today. You are welcome to look at the source code and (if interested) to play with the source itself. However, please do not pass it on.

## 2 A quick tour of the technical details

An electronic structure code requires specific numerical choices that are also reflected in the input files which we will edit further below. These choices are:

### 2.1 Basis set

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

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

There are many possible choices for the shape of the set  $\{\chi_j(\mathbf{r})\}$ : *Gaussian-type orbitals (GTOs)*, so-called *Slater-type orbitals (STOs)*, *plane waves (PWs)*, 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 general shape

$$\chi(\mathbf{r}) = \frac{u(r)}{r} Y_{lm}(\theta, \phi) \quad (4)$$

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 FHI-aims know (via an input file) which of these sets we would like to use.

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.

## 2.2 Numerical integration

For a general single-particle equation

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

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

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

where

$$h_{ij} = \int d^3\mathbf{r}\chi_i^*(\mathbf{r})\hat{h}\chi_j(\mathbf{r}) \quad (7)$$

$$s_{ij} = \int d^3\mathbf{r}\chi_i^*(\mathbf{r})\chi_j(\mathbf{r}). \quad (8)$$

$h_{ij}$  and  $s_{ij}$  are known as the Hamilton and overlap matrices, respectively. Our goal is still to find  $\varphi_m(\mathbf{r})$  and  $\epsilon_m$ , but the problem is now reduced to one of finite-dimensional linear algebra. In practice, it is obviously necessary to perform numerical integrals 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.

## 2.3 Generalized eigenvalue problem

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

## 2.4 Electrostatic (Hartree) potential of the electrons

In all cases,  $\hat{h}$  contains the so-called electrostatic *Hartree potential*  $v_{\text{es}}^e(\mathbf{r})$  of the electron density  $n(\mathbf{r})$ . Finding the Hartree potential is an exercise in solving Poisson’s equation. In brief, FHI-aims uses a decomposition of the Hartree potential in atom-centered multipole components (i.e., a decomposition using spherical harmonics). Do not be surprised if you encounter the maximum expansion order  $l_{\text{hartree}}$ , in the input files.

## 2.5 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, the so-called *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  $\{\varphi_m\}$  (often, based on an initial guess for  $n(\mathbf{r})$ ).
2. Calculate  $n(\mathbf{r})$  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.

For example, we might compare the change of the density from one iteration to the next, the change of the sum of single-particle eigenvalues from one iteration to the next, and the change of the total energy from one iteration to the next. We consider the calculation converged only if all three differences are below a given set of tolerances. Otherwise, we continue the cycle (back to step 2.).

## 3 Single atom calculation with FHI-aims — Step by step

To get comfortable with FHI-aims, we will perform the calculation of the single hydrogen atom in simple steps, explaining what is needed as we go along. The text of this exercise is:

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,
- and that we are interested in results for the *Hartree-Fock* method.

We have to figure out how to run the code in practice, and how to read the output that the code produces so that we can find

- the *single-particle energies* and
- the *total energy*.

All the relevant details for this task are explained below.

### 3.1 Task preparation

The basic protocol is simple:

- Run each calculation in its **own** directory
- There are only two input files, to be edited as ASCII text. They are called:
  - control.in
  - geometry.in

### 3.2 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.

### 3.3 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. Here is a minimal example, which can be used as a template for the exercise:

```
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    50
```

- `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**

Defines the initial spin state of each atom. This keyword is only necessary for spin-polarized calculations. The option `hund` is only suitable for free atoms, where the orbitals are occupied following Hund's rules. **Do not** use the `hund` option for calculations of molecules or solids.

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 unit is eV.

### 3.4 The basis set

The remaining block of `control.in` defines all computational parameters associated with the elements (species) — most importantly, the *basis set*. Each element listed in `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. For each element, these *default species* files can be found in the directory `/programs/FHI-aims/aimsfiles/species_defaults` and should be copied and pasted into `control.in`, e.g. via the command

```
cat /programs/FHI-aims/aimsfiles/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 a given main quantum number  $n$  and an angular momentum quantum number  $l$ . The code then automatically adds the full set of possible basis functions  $\frac{u(r)}{r} Y_{lm}(\theta, \phi)$ , i.e., the functions for all  $m = -l, \dots, +l$ . In

the case given above, all listed basis functions are derived from hydrogen-like central potentials  $-Z/r$ , where  $Z$  is the effective nuclear charge 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* (i.e., levels). Systematically improved calculations can be performed by enabling additional *tiers*.

In addition, the radial functions of the occupied Kohn-Sham eigenstates of spherically symmetric free atoms are always included in the basis set as well. They form the so-called *minimal basis*. Even if all other basis functions are commented out, this choice guarantees that there are always at least as many basis functions available as are needed to represent one orbital per electron (that is, the minimal number of basis functions needed to obtain any result at all).

In the case of hydrogen, the *minimal basis* consists of only one radial function of  $1s$  character. A calculation based only on the minimal basis can be set up by commenting out all other basis functions. In the example above, this means that all the uncommented lines in the *first tier* and *second tier* need to be commented out.

## 4 Finally: The Hydrogen Atom

### 4.1 Energy and Eigenvalues for the Minimal Basis in Hartree-Fock Theory

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, do not forget to set convergence criteria for the SCF cycle, also given above. Finally, paste the *tight* species defaults of H into the `control.in` file, e.g., via the command

```
cat /programs/FHI-aims/aimsfiles/species_defaults/tight/01.H_defaults >> 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:

```
aims.021714.serial.x | tee H.out
```

The `tee` command is a command that will print the output on the screen before you while simultaneously 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 the eigenvalues). How close are they to the result that you expect? What could be the origin of the difference?

## 4.2 Changing to the Local-Density Approximation

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?

## 4.3 Converging the basis set

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 different 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 with the basis set size? Is the variational principle fulfilled?

In general, LDA should not be very good for the total energy of the H atom. This is, of course, not unexpected (it is the worst case for LDA). For instance, the electron interacts with itself if the LDA is used, which is not the case in Hartree-Fock theory.

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.

#### 4.4 Plotting the density

In this exercise, we will examine the total electron density of the hydrogen atom.

We do this using a viewing program for three-dimensional arrays ( $n(\mathbf{r})$  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*.

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

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

to the control.in file.

By default, FHI-aims will figure out the region to plot into the cube files on its own. 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.cube"
```

Next, We should tell *jmol* which *isosurface* of the electron density to display. For example:

```
isosurface cutoff 0.1 "total_density.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 is all you will need to know.

Perform the calculations at the Hartree-Fock level of theory.

## 4.5 Visualizing the radial distribution function

For a spherical atom, there is 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,  $r^2n(r)$ . 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 do so by adding the following lines to the control.in file of the previous task:

```
cube_content_unit      bohr
output  cube total_density
        cube filename total_density_x.cube
        cube origin 0 0 0
        cube edge 9999 0.001 0.0 0.0
        cube edge      1 0.0 0.1 0.0
        cube edge      1 0.0 0.0 0.1
```

`cube origin` defines the origin (centered) of the cube.

`cube edge` defines on how many points along a given edge of our box the density will be written out. The first number indicates how many points are created along one line, while the following three numbers specify the step length along  $x$ ,  $y$ , and  $z$ . Note that each cube file, being a 3-dimensional object, requires 3 edges to be completely defined. With `cube_content_unit bohr` we specify the internal length unit of the cube file. This allows for an easier comparison of the calculated results with the ones obtained in exercise 7.

Based on the resulting `.cube` file, we still need to trace the density along one direction and multiply the density with  $r^2$ . For the setup above, one possibility is to use the python script `cube_trace.py`, which traces the cube file according to the three edge directions for a chosen atomic center. This script produces a two-column file with ASCII data for each direction containing the projected coordinate and the density value. All you need to do is to run the script `cube_trace.py` as follows:

```
cube_trace.py -i total_density.cube
```

The resulting files are `traceX.dat`, `traceY.dat` and `traceZ.dat`. Just use any plotting tool (e.g., `xmgrace`) to plot the ASCII output file you are interested in and perform the multiplication with  $r^2$ . The output units for the radial coordinate are Bohr radii. With this knowledge (the unit of the axis), does the plotted density match your physical intuition?

## 5 The Silicon Atom

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. 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 single-particle eigenvalues and total energies from both calculations. Can you spot the valence and core eigenvalues? How different are they for the two different functionals? Can you rationalize the order of eigenvalues in terms of electron “shells” (note: similar to the analytically known eigenstates of the hydrogen atom as discussed in class a while ago)?
- Plot the three-dimensional electron densities as before. Do they look different?
- Plot the one-dimensional radial distribution function  $r^2n(r)$  for both methods, using the same technique as before. What does it look like? Can you interpret the result?

Comment: If all goes as expected, the final eigenvalues printed for the Si atom will show you so-called “fractional” occupations of the highest-lying eigenstates (0.666667 for the spin-up component).

This is odd, isn't it? What this means is that each  $3p$  eigenstate gets only a fraction of an electron, not a whole electron. How can this be?

In fact, the spherically symmetric state of the Si atom which we have calculated here is “degenerate”: For a single Slater determinant (as in Hartree-Fock theory or in DFT), there are multiple equivalent ways to occupy the highest-lying  $3p$  orbitals with electrons.

In DFT, one way to deal with degenerate states is to use so-called “fractional occupations”. Without going into detail, this is equivalent to a statistical mechanical treatment of the problem, where (in exact DFT) the resulting density would be the averaged density of an ensemble of atoms.

However, the fractionally occupied state is not necessarily the lowest-energy state. We could also try to have the atom *break its spherical symmetry* by selecting only two out of the three possible  $3p$  orbitals to be occupied by a single electron each.

This is not always easy to accomplish. For instance, for Si in the pw-lda density functional, you will not reach a converging solution without further tricks!

However, in Hartree-Fock theory, we can use a simple trick to (effectively) prevent a fractional occupation. Add the following line to your control.in file:

```
occupation_type gaussian 0.000001
```

This line will mandate that only numerically extremely close lying orbitals (within roughly  $10^{-6}$  eV of one another) will have similar occupation numbers, the rest will either be zero or one.

In practice, this change helps the Si atom breaks its symmetry over the course of the self-consistency cycle and fall into a lower-energy non-symmetric solution with integer occupation numbers – but (for the Si atom as treated here) only in Hartree-Fock theory.