2 The Hydrogen Atom: Hartree-Fock Calculations in *Gaussian09*

This set of exercises comprises all the information you will need to run a Hartree-Fock calculation in Gaussian. After giving a detailed account on the most common basis sets, this tutorial introduces you to the world of working in the UNIX command line: You will learn how to navigate, how to create files in a simple text editor (Vi), view, copy and move them; and you shall finally apply this knowledge to your own first electronic structure calculation.

2.1 Basis Sets - Defining Vector Spaces

Three questions have to be addressed before tackling an electronic structure problem: Which computer code is best suited for a given problem, which computational method will give the most accurate results in a reasonable time, and what basis set offers the best compromise of accuracy and efficiency? Throughout this course, you shall always be using the same code (Gaussian09/g09) - but you will get to try out some of the different approaches discussed in the lecture. Before the first practical example - applying the Hartree-Fock-Roothaan scheme (that you have just treated in the lecture) - to the hydrogen atom, there remains one issue to be resolved: What is the basis in which we want to expand our wavefunction that is described by the in principle infinite expansion

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \sum_j c_j \psi_j(\mathbf{r}_1, \ldots, \mathbf{r}_N) \quad ? \tag{66}$$

One-Electron Wavefunctions: Slater-Type Orbitals

By defining a basis set, we define a *vector space* in which the Schrödinger equation is to be solved - and we wish this space to be as close as possible to the complete space that defines the accurate solution. You have already seen that the Hartree-Fock scheme makes a convenient (but not always accurate) approximation to Ψ, in that it is assumed that one Slater determinant is enough to accurately describe the problem. Therefore, in Hartree-Fock theory, the eq. 1 reduces to:

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \psi(\mathbf{r}_1, \ldots, \mathbf{r}_N), \tag{67}$$

where

$$\psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \det |\phi_1(\mathbf{r}_1), \ldots, \phi_N(\mathbf{r}_N)| \tag{68}$$

is a Slater determinant to account for the antisymmetry requirement as discussed in the preceeding chapter, and the $\{\phi\}$ are one-electron orbitals. Although an expression for the many-electron wavefunction in terms of one-particle wavefunctions is now given, the latter are not yet specified. An intuitive approach to the one-electron orbitals may be based on the *LCAO* (Linear Combination of Atomic Orbitals) theory, where one-particle molecular orbitals are formed from one-particle atomic orbitals. This implies

that $\phi_m(\mathbf{r}_m)$ will be expanded in terms of all *atomic one-particle orbitals* of the system, a set of *atomic basis functions*

$$\phi_m(\mathbf{r}_m) = \sum_n D_{mn}\chi_n(\mathbf{r}_m), \tag{69}$$

where the $\{\chi\}$ are the atomic orbitals and D_{mn} is the expansion coefficient (the contribution) of the nth atomic orbital to the single-particle molecular orbital ϕ_m. As the Hartree-Fock many-electron wavefunction is expressed as a single Slater determinant, the coefficients c_j as defined in the introduction vanish, and the only coefficients left in the definition are the D_{mn}. These are the expansion coefficients that are optimised in a Hartree-Fock calculation.

Still, the question how to define the single-particle atomic orbitals is not yet resolved. In principle, the condition that there be a cusp at the nuclei and that the orbital fall off exponentially at large distances from the nuclei dictates a certain form. One suitable form was proposed by Slater in the 30ies of the last century:

$$\chi_{\xi,n,l,m}(\mathbf{r},\theta,\phi) = N \cdot Y_{lm}(\theta,\phi) \cdot r^{n-1} \cdot e^{-\zeta r} \tag{70}$$

A Slater-type orbital is composed of an angular part that is taken from the exact solution of the hydrogen atom Y_{lm} (the spherical harmonics), an exponential part (to ensure the right long-range decay) and a polynomial. However, products of these functions will need to be evaluated - and these are impractically expensive to compute. It is therefore more convenient to choose basis functions that offer some computational advantages. *Gaussian functions* would be especially suited, as products of Gaussians will simply yield another Gaussian that is placed off the initial centres. Frank Boys therefore proposed to approximate Slater-type orbitals with a linear combination of Gaussian-type functions. These Gaussian-type basis functions are referred to as *contraction functions*. This implies that the atomic basis function χ is in turn defined by several basis functions (the term contraction is chosen to avoid confusion between the atomic basis functions, and the linear combination of Gaussians they are based upon):

$$\chi^{STO-3G}_{\xi,n,l,m}(\mathbf{r},\theta,\phi) = \sum_{i=1}^{3} d_i \cdot N_i \cdot Y_{lm}(\theta,\phi) \cdot r^{2n-2-l} \cdot e^{-\xi_i r^2}, \tag{71}$$

where N_i is a normalisation constant, and ξ_i is the ith prefactor in the exponent that guarantees an optimal fit to the Slater-type orbitals. This defines a *minimal Gaussian basis set* known as *STO-3G* (STO stands for Slater-type orbital and refers to the origin of the Gaussian expansion). The term minimal basis does not refer to the number of contractions, but to the number of basis functions: For each orbital, there is one basis function. Minimal bases create minimal computational overhead, but will often not provide sufficient flexibility to accurately describe the system's wavefunction - there is always a certain trade-off between the desired accuracy and the efficiency of a calculation. For more details, you may refer to the main course script.

Pople-Type Split-Valence Basis Sets

Core and valence orbitals are equally important for the energetics of a system, but bonding is dictated by the valence electrons. One may therefore want to improve over the STO-3G basis by allowing for additional flexibility in the description of valence electrons. In a *split-valence basis set*, the number of basis functions that is assigned to core orbitals differs from the one for the valence orbitals. Usually, core electrons are described by one function, which is in turn composed of a certain number of Gaussian functions (*i.e.* contractions). For the description of the valence electrons, multiple functions will be included (most often 2 to 6); and every of these functions will in turn be expressed by a varying number of Gaussian contractions.

An example of a split-valence basis set is John Pople's 3-21G. The notation encodes information about the contraction: The number on the left of the hyphen denotes the number of contractions for the core orbitals, which consist of a single basis function per orbital only. The information on the right describes the contraction of the valence orbitals: There are two numbers, hence there are two basis functions χ per orbital. These basis functions, in turn, are constructed by two and one Gaussian contraction(s) respectively.

Consider, as a practical example, carbon with the electronic configuration $1s^2 2s^2 2p^2$ in the 3-21G basis. The core orbital (1s) is given by a contraction over three Gaussians.

$$\chi(1s) = \sum_{k=1}^{3} \alpha_{1s,k} \mathrm{e}^{-\zeta_{1s,k}\mathbf{r}^2} \tag{72}$$

To every valence orbital (2s and 2p), 1 function containing two Gaussians and one function containing one Gaussian is attributed.

$$\begin{aligned} \chi(2s)^{(2)} &= \sum_{k=1}^{2} \alpha_{2s,k}\ \mathrm{e}^{-\zeta_{2s,k}\mathbf{r}^2} \\ \chi(2s)^{(1)} &= \alpha'_{2s}\ \mathrm{e}^{-\zeta'_{2s}\mathbf{r}^2} \end{aligned} \tag{73}$$

$$\chi(2p)^{(2)} = \sum_{k=1}^{2} \alpha_{2p,k}\ \mathrm{e}^{-\zeta_{2p,k}\mathbf{r}^2}$$
$$\chi(2p)^{(1)} = \alpha'_{2p}\ \mathrm{e}^{-\zeta'_{2p}\mathbf{r}^2} \tag{74}$$

Fixed coefficients are added in front of each Gaussian, denoted by α.
For each atom, there are individual sets of parameters α and ζ, which were determined back when the basis set was designed. These contraction parameters are *never* changed during an electronic structure calculation. Recall that the molecular one-electron wavefunctions are variable linear combinations of *fixed* atomic orbitals; changing the contraction parameters during the calculation would change and therefore mess up the atomic basis functions. The values for standard basis sets are usually hard-coded in the electronic structure codes. For instance, *Gaussian09* represents the basis set parameters in the following format:

```
CARBON     (6S,3P) -> [3S,2P]
S 3
1      172.256000          0.617669000E-01
2      25.9109000          0.358794000
3      5.53335000          0.700713000
L 2
1      3.66498000          -0.395897000          0.23640000
2      0.770545000         1.21584000            0.860619000
L 1
1      0.195857000         1.00000000            1.00000000
```

The S entry contains information about the core, the L entries about the valence orbitals. The first column refers to the index of the contraction k, the second column gives the contraction parameters ζ_k, the third column gives the $\alpha_{s,k}$ and the fourth the $\alpha_{p,k}$. Note that if there is just one contraction, then $\alpha_{l,1} = 1$. In general, s and p orbitals do not differ in ζ_k, but just in $\alpha_{l,k}$.

2.2 Exercises

Please answer the following questions:

2.2.1 A minimal basis set...

a) ...always gives the lowest energy.

b) ...is optimized for small molecules.

c) ...contains one basis function for each atomic orbital only.

2.2.2 A split-valence basis set...

a) ...contains two basis functions for each valence atomic orbital.

b) ...doubles the CPU time of the calculation.

c) ...attributes a different number of basis functions to valence and core orbtials.

2.2.3 Which of the following basis sets does not contain polarisation functions?

a) 6-31G*

b) 6-31G(d,p)

c) 3-21+G

d) DZP

2.2.4 Diffuse functions are added to a basis set to...

a) ...save CPU time.

b) ...better represent electronic effects at larger distances from the nuclei.

c) ...take polarisation into account.

d) ...enhance the description of core orbitals.

2.2.5 Contraction coefficients in the 3-21G basis

Using the information given about the 3-21G contraction coefficients:

a) Give the basis functions corresponding to the 1s, 2s and 2p orbitals of Carbon.

b) If you wish to calculate the Hartree-Fock energy of a carbon atom, how many coefficients are *optimised* during the calculation?

2.2.6 Number of basis functions vs. Gaussian contractions

You wish to calculate the wavefunction of ethylene C_2H_2 using the 6-31G* basis. Indicate the number of basis functions and the number of Gaussian primitives that will be used in the calculation.

2.3 First Steps in *Gaussian09*: The Hydrogen Atom

You will learn how to use *Gaussian09* by putting your hands on a simple example: The total energy of the hydrogen atom in Hartree-Fock theory. This is a tutorial - you are not only invited to type the commands that are being introduced, you are obliged to.

Electronic Structure Software

Ab initio electronic structure software packages make it possible to calculate numerically a variety of properties of a given system, based only on physical constants and the system's Hamiltonian. The only approximations that need to be made are in the method and basis set that have to be chosen, in order to allow for a reasonable computational time. (The stronger your workstation, the more approximations you may drop, and the more elaborate your approach can be.) There are plenty of *ab initio* quantum chemical packages on the market; they differ in their capabilities, license policy and pricing. Widley used packages include GAMESS US, turbomole, DALTON, CP2K, CPMD and the Gaussian set of programs. Although a licence for Gaussian is horrendously expensive, Gaussian is widely distributed because of its ease of use and a reasonable efficiency across a wide range of problems, including:

- Molecular energies and structures
- Bond and reaction energies
- Energies and structures of transition states; transition state search
- Reaction pathways
- Vibrational frequencies
- Molecular orbitals, densities...
- Multipole moments
- Atomic charges and electrostatic potentials
- NMR properties
- ...and many more.

The current version of Gaussian that you will be using is g09; 2009 being the year of the last official release. Gaussian has been developed since the 1970ies by hundreds of researchers, and there is hence a lot of different expertise hidden in the code.

Keyboard instead of double clicks - using a terminal

Unlike most software that you know, Gaussian will not provide you with a user interface; you cannot simply double-click an icon, start the program and make some selections, then hit a 'start' button. Instead, you have to feed Gaussian a pre-prepared input file, and it will in turn spew out an output. You therefore will have to use the *command line* or *terminal* to interact with your computer and with Gaussian. The terminal is simply a way of keyboard-guided interaction with your computer; instead of buttons and clicks, you will have to use commands to make the computer do what you want. (Imagine that, for big supercomputers, there is most often no graphical interface, and everything has to be steered and commanded from a terminal.)

You may open a terminal by using the key combination Ctrl+Alt+T or by clicking on the application launcher (top left icon) and searching for the application "Terminal". Clicking on it will open the command line interface. Before you can start to navigate through your folders in a similar way that you browse through them normally, you need to know about their hierarchical organisation.
The folders are organised in *paths*: If, at home, you have your favourite music in a folder 'Favourites' and this folder sits in a folder called 'Music', this translates to the path `/Music/Favourites` in the command line. The folder that is just designated by `/` is the upper-most folder and is called the root folder. Any path to a folder can either be expressed in terms of the root folder, so you have to give its full path; or you may more simply express it in a relative way by using subdirectories. *I.e* if you are already in the directory 'Music', you can access 'Favourites' by the path `./Favourites`, where the dot tells you that the path is not absolute, but expressed in terms of the current folder. For relative paths, there is also `../`, which refers to a folder that is one folder above the current folder, to return from 'Favourites' to 'Music', you would type `../`.

The following table lists the most important commands. Note that you will have to *replace* the words written in capitals by an actual directory or an actual file name. Every command is sent by hitting `enter` - without hitting `enter`, nothing will happen. All commands are *case sensitive*, also the filenames: a file `File` is not the same as a file called `file`.

Basic UNIX Commands

- `pwd` outputs the path to the directory you are currently working in
- `ls DIRECTORY` lists the contents of the directory named `DIRECTORY`. Omitting `DIRECTORY` will list the contents of the current directory.
- `ls -lrsth DIRECTORY` dito, but with more detail and colour-coding
- `ls ./` lists the content of the current directory
- `cd DIRECTORY` changes to the directory named `DIRECTORY`
- `cd ../` changes one directory up
- `cp DIRECTORY/FILE DIRECTORY2/FILE2` copies the file `FILE` in `DIRECTORY` to a file name `FILE2` in directory `DIRECTORY2`
- `mv DIRECTORY/FILE DIRECTORY2/FILE2` dito, but moving instead of copying
- `rm FILE` removes the file named `FILE`. Note that the file does *not* go to the trash, but is deleted irreversibly.
- `rmdir DIRECTORY` removes the directory named `DIRECTORY`
- `mkdir DIRECTORY` creates a directory named `DIRECTORY`
- `cat FILE` displays all the content of the file `FILE` on the screen
- `grep 'WHAT_YOU_ARE_SEARCHING_FOR' FILE` extracts a line from the file `FILE` that contains the word `WHAT_YOU_ARE_SEARCHING_FOR`
- `less FILE` displays the content of the file `FILE`, and allows to navigate
- `vi FILE` opens vi, a program to view and edit files; here, the file is called `FILE`.
- `Tab` allows for automatic expansion: If you type `ls` (or `cp`, `mv`...) followed by the `Tab` key, all files in the current working directory, as well as all subdirectories, will be listed. If you have already typed one or several letters before hitting `Tab`, only subdirectories and files beginning with these letters will be displayed. If there is only one file or subdirectory that matches the pattern, `Tab` automatically expands the full file name.
- ↑ `(arrow up)` and ↓ `(arrow down)` allow you to scroll through a list of your last commands. Use ↑ to go back in the history, and ↓ to move to more recent commands again.

The following tutorial will give you an example on how to use the commands. Type

```
ls /
```

which will display the contents of the root directory. Gaussian is located in the path `/usr/local/gaussian/g09-D.01/g09/`, so by typing

```
ls /usr/local/gaussian/g09-D.01/g09/
```

you will see all the files and directories that are in the g09 directory. Now, change to this directory; but instead of writing the full path, try using the `Tab` expansion:

```
cd /u
```

and, instead of hitting enter, hit `Tab`. This should expand to

```
cd /usr/
```

and by hitting `Tab` again, all subdirectories and files within `/usr/` will be displayed. Type a `l` after the path,

```
cd /usr/l
```

which, after using `Tab`, will expand to `/usr/local/`, and then hit enter. Display the path to your current working directory by

```
pwd
```

and list its content again, this time using the more verbose output:

```
ls -lrsth
```

Now, you may return to the root directory by typing either:

```
cd /
```

or, with relative paths,

```
cd ../../
```

You have now succesfully navigated through folders using the terminal.

A Simple Text Editor: Vi

Most quantum chemical programs will only accept files that contain *plain text*. You cannot create such a file using, for instance, Microsoft Word, as this will generate a file that contains much more data than just the text you typed (all the formatting information will also be contained in the file itself !). Hence, one has to use an ASCII editor that is capable of creating simple plain-text files. Vi is one of the best choices for this. From the command line, you may open a file called `my_test` by typing

```
vi my_test
```

which will also create the file `my_test` if it doesn't exist yet. After hitting enter, you will find yourself in vi. Vi, as a particularity, has three different working modes:

- Normal mode: If you start the program or if you hit the `Esc` button. In this mode, you can give commands to vi, as specified later.
- Insert mode: By either typing `i` or hitting the `Insert` key, you will be in insert mode. If you type something now, it will be inserted in your file.
- Replace mode: By hitting `Insert` again in the insert mode, you will pass to replace mode. If you start typing, vi will replace the letters below your cursor with a new one, rather than adding them. By hitting `Insert` again, you switch back to regular insert mode.

In your freshly opened file, switch to insert mode by pressing `i` or `Insert`. Then, type something you like, such as 'Vi rocks.' (Hit insert again and type something to see what happens if you pass to replace mode.) To save the file, you have to exit the insert mode by pressing `Esc`. In escape mode, type `:w` to save the file, or type `:w filename` to save it under a file called `filename` rather than `my_test` as initially specified. Before exiting vi, type `/rocks` in normal mode (use a word that you wrote in the document): The slash works as a search tool for strings, and you should now see the word 'rocks' marked in your file (or whatever word you chose. Note that if there are multiple matches in your file, you can jump between them using the `n` key). You may exit vi by typing `:q`. If there were changes since your last save that you would want to keep, type `:wq`; if you want to discard changes since the last save, type `:q!`, where the ! forces vi to quit. In a related matter, to overwrite an existing file, use `:w!` instead of `:w`.

Writing An Input For *Gaussian09* and Invoking the Program

Using your newly acquired knowledge on vi, create a file `H_STO-3G.com` that contains the following information. However, exclude the line numbers on the left - they are just there for the following explanation.

```
1    %NProcShared=2
2    #P UHF/STO-3G SP
3
4    H-atom single point
5
6    0 2
7    H 0.0000 0.0000 0.0000
8
```

At the end, your input *must* contain a blank line, or Gaussian will encounter an error reading it. This is a relict of the old code frome the 1970ies.
Gaussian input is always structured in the same way:

- 1 - The first line tells Gaussian to use both processors of your workstation, rather than just one.
- 2 - The *route section* specifies the method and the basis set, separated by a forward slash. After a space, the type of calculation has to be specified (SP = single point wavefunction optimisation). `#P` asks for verbose output.
- 3 - There must be a blank line.
- 4 - This is the title of your calculation, you may type anything that tells you what it is about.
- 5 - There must be a blank line. Again.
- 6 - The total charge of the system, followed by the spin multiplicity.
- 7 - All the atoms of your system with their x,y,z coordinate. One atom per line.
- 8 - There must be a blank line. This tells Gaussian that the input has ended and that there are no more new atoms to add.

Gaussian will now read everything it needs from this input file: The (initial) molecular structure, the method, the basis set, the charge and the multiplicity.

Create two more inputs called `H_6-31G.com` and `H_6-311G.com`, where you replace the `STO-3G` basis set instruction by the ones for the `6-31G` and `6-311G` basis respectively. Now, for the STO input file, invoke Gaussian:

```
g09 < H_STO-3G.com > H_STO-3G.log
```

If the g09 command is not found by the computer, complete the command with the absolute path to the g09 executable,

```
/usr/local/gaussian/g09-D.01/g09/g09 < H_STO-3G.com > H_STO-3G.log
```

The > tells Gaussian to redirect its output to a file called `H_STO-3G.log`, rather than the command line (where the output would be lost once you close the terminal). You will not be able to type anything until Gaussian has finished - this is the usual behaviour when invoking a program. As soon as the calculation has finished, you will be able to type normally again.

Viewing the Output

List the files in your directory (`ls`) to see what new files have been generated. Indeed, you should find the `.log` file created by Gaussian. As this file may be very large and we have no intention of editing it, we may use `less` to display it. Type `less` followed by the file name: You may 'jump' through the file using the enter key, or you may directly drop to its end using a capital `G` (*i.e.* shift and g). Pop back to the top by typing a lowercase `g`. For a more finely tuned navigation, simply resort to the arrow keys. Just like in vi, typing `/` followed by a word will jump to said word.
First, go to the end of the file to see whether Gaussian has exited without causing problems (the last line should read `Normal termination` - if `Error termination` appears, something has gone wrong). The end of the file contains a rather cryptic block called the *archive entry* and a final messages telling you whether Gaussian has exited normally. There is also a fortune cookie in the end, which may be a silly scientific joke, a quotation, or just anything that has been haunting the eads of the people who wrote the Gaussian code.

```
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l9999.exe)
1\1\GINC-LCBCPC41\SP\UHF\STO-3G\H1(2)\MARTIN\23-Sep-2014\0\\#P UHF/STO
-3G SP\\H-atom single point calculation\\0,2\H,0,0.,0.,0.\\Version=ES6
4L-G09RevD.01\State=2-A1G\HF=-0.4665819\S2=0.75\S2-1=0.\S2A=0.75\RMSD=
0.000e+00\Dipole=0.,0.,0.\Quadrupole=0.,0.,0.,0.,0.,0.\PG=OH [O(H1)]\\
@


IF MATHEMATICALLY YOU END UP WITH THE INCORRECT ANSWER,
TRY MULTIPLYING BY THE PAGE NUMBER.
Job cpu time:       0 days  0 hours  0 minutes  1.6 seconds.
File lengths (MBytes):  RWF=     67 Int=      0 D2E=      0 Chk=      2 Scr=      2
Normal termination of Gaussian 09 at Tue Sep 23 14:34:03 2014.
```

If Gaussian has terminated without an error, go back to the top of the file to examine it as a whole. The first part of the output is general information about the program and licensing. It also includes the appropriate citation, in case you publish any results obtained by using Gaussian.

```
Entering Gaussian System, Link 0=g09
Initial command:
/software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l1.exe "/share/lcbcpc34/data1/user21/martin/ENSEIGNEME
Entering Link 1 = /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l1.exe PID=       329.

Copyright (c) 1988,1990,1992,1993,1995,1998,2003,2009,2013,
           Gaussian, Inc.  All Rights Reserved.

This is part of the Gaussian(R) 09 program.  It is based on
the Gaussian(R) 03 system (copyright 2003, Gaussian, Inc.),
the Gaussian(R) 98 system (copyright 1998, Gaussian, Inc.),
the Gaussian(R) 94 system (copyright 1995, Gaussian, Inc.),
the Gaussian 92(TM) system (copyright 1992, Gaussian, Inc.),
the Gaussian 90(TM) system (copyright 1990, Gaussian, Inc.),
the Gaussian 88(TM) system (copyright 1988, Gaussian, Inc.),
the Gaussian 86(TM) system (copyright 1986, Carnegie Mellon
University), and the Gaussian 82(TM) system (copyright 1983,
Carnegie Mellon University). Gaussian is a federally registered
trademark of Gaussian, Inc.

This software contains proprietary and confidential information,
including trade secrets, belonging to Gaussian, Inc.

This software is provided under written license and may be
used, copied, transmitted, or stored only in accord with that
written license.

The following legend is applicable only to US Government
contracts under FAR:

                    RESTRICTED RIGHTS LEGEND

Use, reproduction and disclosure by the US Government is
subject to restrictions as set forth in subparagraphs (a)
and (c) of the Commercial Computer Software - Restricted
Rights clause in FAR 52.227-19.

Gaussian, Inc.
340 Quinnipiac St., Bldg. 40, Wallingford CT 06492


---------------------------------------------------------------
Warning -- This program may not be used in any manner that
competes with the business of Gaussian, Inc. or will provide
assistance to any competitor of Gaussian, Inc.  The licensee
of this program is prohibited from giving any competitor of
Gaussian, Inc. access to this program.  By using this program,
the user acknowledges that Gaussian, Inc. is engaged in the
business of creating and licensing software in the field of
computational chemistry and represents and warrants to the
licensee that it is not a competitor of Gaussian, Inc. and that
it will not use this program in any manner prohibited above.
---------------------------------------------------------------
```

```
Cite this work as:
Gaussian 09, Revision D.01,
M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria,
M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci,
G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian,
A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada,
M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima,
Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr.,
J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers,
K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand,
K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi,
M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross,
V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann,
O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski,
R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth,
P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels,
O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski,
and D. J. Fox, Gaussian, Inc., Wallingford CT, 2013.
```

The route section is then displayed as read from the input, followed by some internal parameters (there is a huge documentation available online, explaining what they mean).

```
******************************************
Gaussian 09:  ES64L-G09RevD.01 24-Apr-2013
              23-Sep-2014
******************************************
%nproc=2
Will use up to    2 processors via shared memory.
----------------
#P UHF/STO-3G SP
----------------
1/38=1/1;
2/12=2,17=6,18=5,40=1/2;
3/6=3,11=2,16=1,25=1,30=1,116=2/1,2,3;
4//1;
5/5=2,38=5/2;
6/7=2,8=2,9=2,10=2,28=1/1;
99/5=1,9=1/99;
Leave Link    1 at Tue Sep 23 14:34:01 2014, MaxMem=           0 cpu:         0.1
```

This is followed by both the input and standard orientation of your atom or molecule. In the case of a single atom, both are the same. In general, however, they will differ; the standard orientation being an internal orientation by which Gaussian optimises computational efficiency. The title card is also repeated here.

```
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l101.exe)
------------------------------
H-atom single point calculation
------------------------------
Symbolic Z-matrix:
Charge =  0 Multiplicity = 2
H                     0.        0.        0.

NAtoms=      1 NQM=        1 NQMF=       0 NMMI=      0 NMMIF=      0
               NMic=       0 NMicF=      0.
                    Isotopes and Nuclear Properties:
(Nuclear quadrupole moments (NQMom) in fm**2, nuclear magnetic moments (NMagM)
 in nuclear magnetons)

 Atom         1
IAtWgt=           1
AtmWgt=   1.0078250
NucSpn=           1
AtZEff=   0.0000000
NQMom=    0.0000000
NMagM=    2.7928460
AtZNuc=   1.0000000
Leave Link  101 at Tue Sep 23 14:34:01 2014, MaxMem=    33554432 cpu:         0.2
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l202.exe)
                         Input orientation:
---------------------------------------------------------------------
Center     Atomic      Atomic             Coordinates (Angstroms)
Number     Number       Type             X           Y           Z
---------------------------------------------------------------------
     1          1           0        0.000000    0.000000    0.000000
---------------------------------------------------------------------
Stoichiometry    H(2)
Framework group  OH[O(H)]
Deg. of freedom     0
Full point group                 OH      NOp  48
Largest Abelian subgroup         D2H     NOp   8
Largest concise Abelian subgroup C1      NOp   1
                         Standard orientation:
---------------------------------------------------------------------
Center     Atomic      Atomic             Coordinates (Angstroms)
Number     Number       Type             X           Y           Z
---------------------------------------------------------------------
     1          1           0        0.000000    0.000000    0.000000
---------------------------------------------------------------------
Leave Link  202 at Tue Sep 23 14:34:01 2014, MaxMem=    33554432 cpu:         0.0
```

This is followed by information about the chosen basis functions and their symmetry.

```
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l301.exe)
Standard basis: STO-3G (5D, 7F)
```

```
Ernie: Thresh=  0.10000D-02 Tol=  0.10000D-05 Strict=F.
There are     1 symmetry adapted cartesian basis functions of AG  symmetry.
There are     0 symmetry adapted cartesian basis functions of B1G symmetry.
There are     0 symmetry adapted cartesian basis functions of B2G symmetry.
There are     0 symmetry adapted cartesian basis functions of B3G symmetry.
There are     0 symmetry adapted cartesian basis functions of AU  symmetry.
There are     0 symmetry adapted cartesian basis functions of B1U symmetry.
There are     0 symmetry adapted cartesian basis functions of B2U symmetry.
There are     0 symmetry adapted cartesian basis functions of B3U symmetry.
There are     1 symmetry adapted basis functions of AG  symmetry.
There are     0 symmetry adapted basis functions of B1G symmetry.
There are     0 symmetry adapted basis functions of B2G symmetry.
There are     0 symmetry adapted basis functions of B3G symmetry.
There are     0 symmetry adapted basis functions of AU  symmetry.
There are     0 symmetry adapted basis functions of B1U symmetry.
There are     0 symmetry adapted basis functions of B2U symmetry.
There are     0 symmetry adapted basis functions of B3U symmetry.
    1 basis functions,     3 primitive gaussians,     1 cartesian basis functions
    1 alpha electrons        0 beta electrons
      nuclear repulsion energy         0.0000000000 Hartrees.
IExCor=    0 DFT=F Ex=HF Corr=None ExCW=0 ScaHFX=  1.000000
ScaDFX=  1.000000  1.000000  1.000000  1.000000 ScalE2=  1.000000  1.000000
IRadAn=      0 IRanWt=     -1 IRanGd=            0 ICorTp=0 IEmpDi=  4
NAtoms=    1 NActive=    1 NUniq=    1 SFac= 1.00D+00 NAtFMM=   60 NAOKFM=F Big=F
Integral buffers will be    131072 words long.
Raffenetti 2 integral format.
Two-electron integral symmetry is turned on.
Leave Link  301 at Tue Sep 23 14:34:02 2014, MaxMem=    33554432 cpu:         0.1
```

Then follows the main computational block, where the output may help you keep track of what Gaussian is (or is not) doing.

```
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l302.exe)
NPDir=0 NMtPBC=     1 NCelOv=     1 NCel=       1 NClECP=     1 NCelD=      1
        NCelK=      1 NCelE2=     1 NClLst=     1 CellRange=     0.0.
One-electron integrals computed using PRISM.
NBasis=     1 RedAO= T EigKep=  2.83D+00  NBF=     1     0     0     0     0     0     0     0
NBsUse=     1 1.00D-06 EigRej= -1.00D+00 NBFU=     1     0     0     0     0     0     0     0
Leave Link  302 at Tue Sep 23 14:34:02 2014, MaxMem=    33554432 cpu:         0.2
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l303.exe)
DipDrv:  MaxL=1.
Leave Link  303 at Tue Sep 23 14:34:02 2014, MaxMem=    33554432 cpu:         0.1
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l401.exe)
ExpMin= 1.69D-01 ExpMax= 3.43D+00 ExpMxC= 3.43D+00 IAcc=1 IRadAn=         1 AccDes= 0.00D+00
Harris functional with IExCor=  205 and IRadAn=       1 diagonalized for initial guess.
HarFok:  IExCor=  205 AccDes= 0.00D+00 IRadAn=         1 IDoV= 1 UseB2=F ITyADJ=14
ICtDFT=  3500011 ScaDFX=  1.000000  1.000000  1.000000  1.000000
FoFCou: FMM=F IPFlag=           0 FMFlag=      100000 FMFlg1=           0
        NFxFlg=           0 DoJE=T BraDBF=F KetDBF=T FulRan=T
```

```
        wScrn=  0.000000 ICntrl=     500 IOpCl=  0 I1Cent=   200000004 NGrid=           0
        NMat0=    1 NMatS0=      1 NMatT0=    0 NMatD0=    1 NMtDS0=    0 NMtDT0=    0
Petite list used in FoFCou.
Harris En=-0.409939313200721
JPrj=0 DoOrth=F DoCkMO=F.
Initial guess orbital symmetries:
Alpha Orbitals:
      Occupied  (A1G)
Beta  Orbitals:
      Virtual   (A1G)
The electronic state of the initial guess is 2-A1G.
Initial guess <Sx>= 0.0000 <Sy>= 0.0000 <Sz>= 0.5000 <S**2>= 0.7500 S= 0.5000
Leave Link  401 at Tue Sep 23 14:34:02 2014, MaxMem=    33554432 cpu:         0.2
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l502.exe)
UHF open shell SCF:
Using DIIS extrapolation, IDIIS=  1040.
Integral symmetry usage will be decided dynamically.
Keep R1 and R2 ints in memory in symmetry-blocked form, NReq=820858.
IVT=       20204 IEndB=       20204 NGot=    33554432 MDV=    33534121
LenX=    33534121 LenY=    33533239
Requested convergence on RMS density matrix=1.00D-08 within 128 cycles.
Requested convergence on MAX density matrix=1.00D-06.
Requested convergence on             energy=1.00D-06.
No special actions if energy rises.
FoFCou: FMM=F IPFlag=           0 FMFlag=           0 FMFlg1=           0
        NFxFlg=           0 DoJE=F BraDBF=F KetDBF=F FulRan=T
        wScrn=  0.000000 ICntrl=     600 IOpCl=  0 I1Cent=           0 NGrid=           0
        NMat0=    1 NMatS0=      1 NMatT0=    0 NMatD0=    1 NMtDS0=    0 NMtDT0=    0
Petite list used in FoFCou.

Cycle   1  Pass 1  IDiag  1:
E=-0.466581850384435
DIIS: error= 0.00D+00 at cycle   1 NSaved=   1.
NSaved= 1 IEnMin= 1 EnMin=-0.466581850384435     IErMin= 1 ErrMin= 0.00D+00
ErrMax= 0.00D+00  0.00D+00 EMaxC= 1.00D-01 BMatC= 0.00D+00 BMatP= 0.00D+00
IDIUse=1 WtCom= 1.00D+00 WtEn= 0.00D+00
Coeff-Com:  0.100D+01
Coeff:      0.100D+01
Skip diagonalization as Alpha Fock matrix is already diagonal.
Skip diagonalization as Beta Fock matrix is already diagonal.
RMSDP=0.00D+00 MaxDP=0.00D+00              OVMax= 0.00D+00

SCF Done:  E(UHF) = -0.466581850384     A.U. after    1 cycles
           NFock=  1  Conv=0.00D+00     -V/T= 1.6139
<Sx>= 0.0000 <Sy>= 0.0000 <Sz>= 0.5000 <S**2>= 0.7500 S= 0.5000
<L.S>= 0.000000000000E+00
KE= 7.600318798919D-01 PE=-1.226613730276D+00 EE= 0.000000000000D+00
Annihilation of the first spin contaminant:
S**2 before annihilation     0.7500,   after     0.7500
Leave Link  502 at Tue Sep 23 14:34:02 2014, MaxMem=    33554432 cpu:         0.1
```

The most important part of this block is the `SCF done` statement, where the optimised

energy for the given geometry is printed. Note that energies are always given in atomic units (Hartree; 1 a.u. = 627.5 kcal·mol^{-1}).

```
SCF Done:  E(UHF) = -0.466581850384     A.U. after    1 cycles
```

The next block prints the final information that is obtained from the optimised wavefunction: Dipole moments, population analysis and more.

```
(Enter /software/gaussian/G09-D.01/Intel_SSE4.2_support/g09/l601.exe)
Copying SCF densities to generalized density rwf, IOpCl= 1 IROHF=0.

******************************************************************

           Population analysis using the SCF density.

******************************************************************

Orbital symmetries:
Alpha Orbitals:
      Occupied  (A1G)
Beta  Orbitals:
      Virtual   (A1G)
The electronic state is 2-A1G.
Alpha  occ. eigenvalues --   -0.46658
 Beta virt. eigenvalues --    0.30802
         Condensed to atoms (all electrons):
              1
    1  H    1.000000
         Atomic-Atomic Spin Densities.
              1
    1  H    1.000000
Mulliken charges and spin densities:
              1          2
    1  H    0.000000   1.000000
Sum of Mulliken charges =   0.00000   1.00000
Mulliken charges and spin densities with hydrogens summed into heavy atoms:
              1          2
Electronic spatial extent (au):  <R**2>=              1.9486
Charge=              0.0000 electrons
Dipole moment (field-independent basis, Debye):
   X=              0.0000    Y=              0.0000    Z=              0.0000  Tot=              0.0000
Quadrupole moment (field-independent basis, Debye-Ang):
  XX=             -0.8736   YY=             -0.8736   ZZ=             -0.8736
  XY=              0.0000   XZ=              0.0000   YZ=              0.0000
Traceless Quadrupole moment (field-independent basis, Debye-Ang):
  XX=              0.0000   YY=              0.0000   ZZ=              0.0000
```

```
  XY=              0.0000   XZ=              0.0000   YZ=              0.0000
Octapole moment (field-independent basis, Debye-Ang**2):
 XXX=              0.0000  YYY=              0.0000  ZZZ=              0.0000  XYY=              0.0000
 XXY=              0.0000  XXZ=              0.0000  XZZ=              0.0000  YZZ=              0.0000
 YYZ=              0.0000  XYZ=              0.0000
Hexadecapole moment (field-independent basis, Debye-Ang**3):
XXXX=             -0.7029 YYYY=             -0.7029 ZZZZ=             -0.7029 XXXY=              0.0000
XXXZ=              0.0000 YYYX=              0.0000 YYYZ=              0.0000 ZZZX=              0.0000
ZZZY=              0.0000 XXYY=             -0.2343 XXZZ=             -0.2343 YYZZ=             -0.2343
XXYZ=              0.0000 YYXZ=              0.0000 ZZXY=              0.0000
N-N= 0.000000000000D+00 E-N=-1.226613730276D+00  KE= 7.600318798919D-01
Symmetry AG   KE= 7.600318798919D-01
Symmetry B1G  KE= 0.000000000000D+00
Symmetry B2G  KE= 0.000000000000D+00
Symmetry B3G  KE= 0.000000000000D+00
Symmetry AU   KE= 0.000000000000D+00
Symmetry B1U  KE= 0.000000000000D+00
Symmetry B2U  KE= 0.000000000000D+00
Symmetry B3U  KE= 0.000000000000D+00
                         Isotropic Fermi Contact Couplings
       Atom                 a.u.       MegaHertz       Gauss      10(-4) cm-1
    1  H(1)               0.39469    1764.23530     629.52300     588.48555
---------------------------------------------------------------
      Center         ----  Spin Dipole Couplings  ----
                     3XX-RR        3YY-RR        3ZZ-RR
---------------------------------------------------------------
    1   Atom        0.000000      0.000000      0.000000
---------------------------------------------------------------
                       XY            XZ            YZ
---------------------------------------------------------------
    1   Atom        0.000000      0.000000      0.000000
---------------------------------------------------------------


--------------------------------------------------------------------------------
             Anisotropic Spin Dipole Couplings in Principal Axis System
--------------------------------------------------------------------------------

      Atom             a.u.   MegaHertz   Gauss  10(-4) cm-1        Axes

             Baa     0.0000     0.000     0.000     0.000  1.0000  0.0000  0.0000
    1 H(1)   Bbb     0.0000     0.000     0.000     0.000  0.0000  1.0000  0.0000
             Bcc     0.0000     0.000     0.000     0.000  0.0000  0.0000  1.0000


--------------------------------------------------------------------------------

No NMR shielding tensors so no spin-rotation constants.
Leave Link  601 at Tue Sep 23 14:34:03 2014, MaxMem=    33554432 cpu:         0.3
```

Then follows the archive entry that we have already discussed in the beginning. Having scrolled through all the output, you can quit `less` by typing `q` (again, just as in vi).

2.4 Practical Exercises

2.4.1 Effects Of Basis Set Size

You have already prepared two more input files besides the `H_STO-3G.com`, namely the ones for the 6-31G and 6-311G basis. As for STO-3G, calculate the energies of the H atom in these bases. Please note that, in the Gaussian command, you will have to name the output file appropriately, so that `H_STO-3G.com` is not overwritten! (As a reminder, the output file is the one that follows the `>`.)

a) Complete the following table, including the number of basis functions used. The exact energy for the H atom is given by the analytical expression:

$$E = \frac{1}{2} m_e c^2 \alpha^2, \tag{75}$$

where α is the fine structure constant.

$$m_e = 0.910953 \cdot 10^{-30} kg \tag{76}$$

$$c = 2.99792458 \cdot 10^{8} m s^{-1} \tag{77}$$

$$\alpha = 7.2973525376 \cdot 10^{-3} \tag{78}$$

$$N_A = 6.0221367 \cdot 10^{23} mol^{-1} \tag{79}$$

Pay attention to the units - use atomic units or kcal·mol^{-1} throughout (also in the table).

Method	Number of basis functions	Total Energy [a.u.]
UHF/STO-3G		
UHF/6-31G		
UHF/6-311G		
Exact value	(analytical)	

b) What is the difference between the exact number and the one calculated with UHF/STO-3G?

c) What is the influence of the basis set size on the accuracy of the result? How do the split-valence bases compare to STO-3G?

2.4.2 RHF vs. UHF

This exercise is already a preparation for the next set of exercises - you will now calculate a molecular structure, rather than an isolated atom. Using the same basis throughout (6-31G), you should compare H_2 at equilibrium distance (0.74 Å) and at a larger distance of 5.6 Å at two different levels of Hartree-Fock: Restricted Hartree-Fock (RHF) and Unrestricted Hartree-Fock (UHF). Copy the 6-31G input file to:

```
cp H_631G.com H2_631G_UHF_eq.com
```

Open the new file called `H2_631G_UHF_eq.com` in vi and add another line under the first coordinate:

```
H 0.0000 0.0000 0.7414
```

In the route section, add the term `Guess=Mix` and change the multiplicity from 2 to 1. Do not forget about the final blank line! Your new input file will look like this:

```
%NProcShared=2
#P UHF/6-31G SP Guess=Mix

H2 UHF 0.74 A

0 1
H 0.0000 0.0000 0.0000
H 0.0000 0.0000 0.7414
```

Copy this file again (you may name the new file `H2_631G_UHF_large.com`), and replace the z coordinate `0.7414` by `5.6000`. You now have two molecular inputs in UHF, at equilibrium and large distances. Now, copy these files again to files that you name `RHF` instead of `UHF`, and in the input, replace the keyword UHF by RHF. (You may also remove the `Guess=Mix` option, as this will be ignored in a RHF run. However, Gaussian will automatically skip this keyword, so you could leave it in. We however recommend to keep input files as clean as possible and to avoid any unused keywords.) You should now have a collection of four input files:

```
H2_631G_UHF_eq.com
H2_631G_UHF_large.com
H2_631G_RHF_eq.com
H2_631G_RHF_large.com
```

You may, of course, name them differently, but please ensure that you do not overwrite your old in- and outputs, *i.e.* for each new calculation, create always new input and output files! Invoke Gaussian with these input files (you may again denote the output files with the same name as the input, but with the extension `.log` instead of `.com`), *e.g.*

```
g09 < H2_631G_RHF_large.com > H2_631G_RHF_large.log &
```

The ampersand `&` at the end will submit the calculation in the background, so that you can immediately continue your work in the terminal. If Gaussian has terminated normally, you may quickly extract the energy using the `grep` command:

```
grep 'SCF Done' H2_631G_RHF_large.log
```

which will display the line containing `SCF Done` on the screen. Do this for every output file.

Distance [Å]	E_{RHF} [a.u.]	E_{UHF} [a.u.]	$E_{UHF-RHF}$ [a.u.]	$E_{UHF-RHF}$ [kcal· mol^{-1}]
0.74				
5.60				

a) Give the energies and the difference in energies ($E_{UHF} - E_{RHF}$) for both bond distances (*cf.* the following table).

b) The energy to break a chemical bond is usually between 20 and 100 kcal/mol. Explain the roots and physical origin of the difference $E_{UHF-RHF}$ (in your own words). Why is the energy gap between UHF and RHF larger at a larger bond distance?

2.5 Theoretical Exercises

2.5.1 The Output

a) What is the significance of the statement `SCF Done`?

b) Why is `SCF Done` followed by `after n cycles`? Compare the number of cycles for the different basis sets.

c) Why do you have to change the spin multiplicity when moving from an atom to a molecule? How do you calculate the spin multiplicity of a species?