

# Getting Started – Day 1

**Tutors:** Fernanda Duarte (fernanda.duartegonzalez@chem.ox.ac.uk)  
Henry Chan (ho.chan@chem.ox.ac.uk)  
Alistair Sterling (alistair.sterling@chem.ox.ac.uk)

The most updated version of this handout can always be found at:

<https://github.com/duartegroup/sbmcc/blob/master/sbmcc.pdf>

Welcome! In this set of exercises you will gain some hands-on experience in connecting to a remote cluster, creating input files for the computational chemistry programme ORCA, and submitting the calculations to the cluster's job management system.

Before starting, please install **MobaXterm** (from the Software Centre), **Avogadro** (version 1.2.0 from <https://avogadro.cc/>; the version in the Software Centre, 1.1.1, is not sufficient). You will need both these programmes to complete the following exercises. We also recommend installing a plain text editor such as **Notepad++** (<https://notepad-plus-plus.org/>).

As we go along, there will be certain sections where you are asked to note down a value or to answer some questions. **You will need to submit a short writeup (approximately one page long) containing these at the end of the course.**

We will not discuss the theory behind the calculations: if you are interested, a few suggestions for books can be found in the *Other resources* section at the end of this handout.

## 1 Connecting to the cluster

The cluster we will be using is part of the Advanced Research Computing (ARC) facilities at Oxford (<https://help.it.ox.ac.uk/arc/index>). We need to connect to it remotely (via SSH), a functionality which is provided by MobaXterm. When opening MobaXterm for the first time, you may encounter a Windows Firewall prompt: make sure that access over domain networks is allowed.

Click **Session** (not “Sessions”!) at the top-left corner, then **SSH**. In the resulting popup window (*Figure 1*), enter `arcus-b.arc.ox.ac.uk` in the *Remote host* box, tick the *Specify username* checkbox, and enter the username you were given. Leave the remainder of the settings unchanged.

You can skip this in subsequent sessions by using the *Remote sessions* section in the MobaXterm start screen.

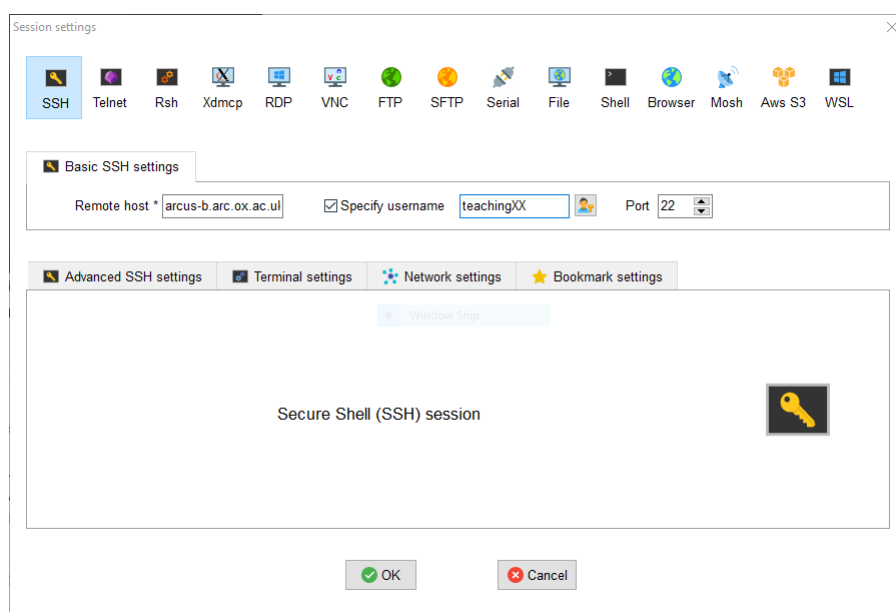


Figure 1: The SSH dialog in MobaXterm.

Click **OK**. Enter your password when prompted, then press Enter. You should be greeted with a terminal window on the right, and a graphical interface allowing you to browse files on the left (*Figure 2*):

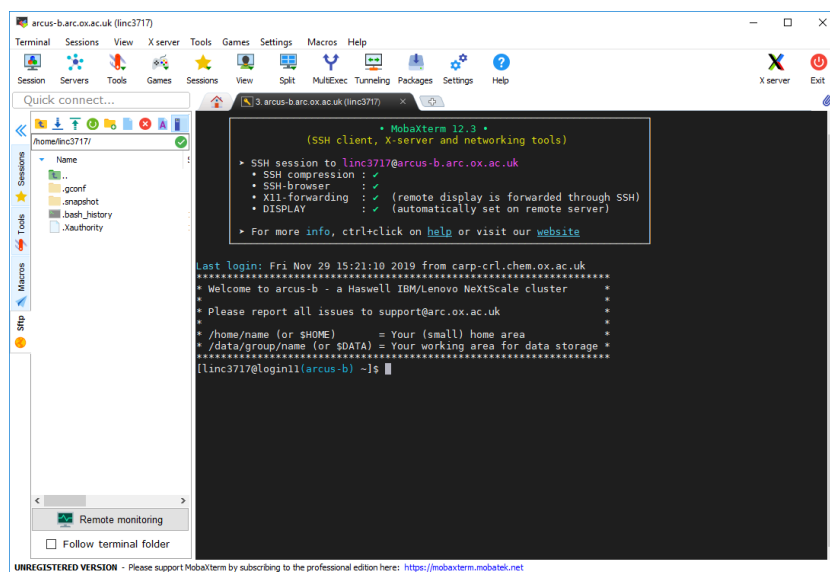


Figure 2: The MobaXterm interface upon successfully logging in.

Congratulations! You are now logged into the ARC cluster: it is from this screen that you will be able to submit calculations and monitor their output.

## 2 Managing files on the cluster

The left-hand pane in MobaXterm resembles the “familiar” File Explorer in Windows. Perhaps most importantly, it allows you to move files to and from the cluster in a drag-and-drop manner. You can try it with any file from your computer (but bear in mind that the file transfer can take a while depending on the file size).

Although this feature often comes in handy, it is also important to be (somewhat) comfortable with using the terminal. In this section we will go through several of the most basic commands for file and folder management. You may wish to skip ahead if you are already familiar with these.

The first thing that we notice is that the line you input text on is preceded by a series of characters (called a *prompt*):

```
[username@hostname(arc-login02) ~]$
```

The prompt contains some important information: your username, the hostname (the name of the computer you are connected to), and the current *directory* you are in. When you first log in, you will always find yourself in your *home directory*, which is conventionally denoted by a tilde ~. The command

```
ls
```

can be used to list the contents of the directory you are in. If you try that right now, you will probably get no output, because your home directory should be empty. We can populate it with a new (empty) folder by using:

```
mkdir cats
```

and a new file with:

When entering passwords into terminals, there is no visual feedback, so don't be alarmed.

Linux, MacOS, and Windows 10 systems have *native* SSH capabilities, meaning that in principle you do not need to install any extra software to connect to a remote server. However, they do not provide the graphical interface in the left side of MobaXterm, so everything must be done exclusively from the terminal.

The dollar sign \$ has no particular meaning; it just happens to be one of the most commonly used characters for ending a prompt.

```
touch maine_coon.txt
```

Now running `ls` should tell us that there are two things in the current directory, namely `cats` and `maine_coon.txt` (try it!). This is not immediately reflected in the graphical browser; you will need to click on the *Refresh* button (Figure 3) to see them:

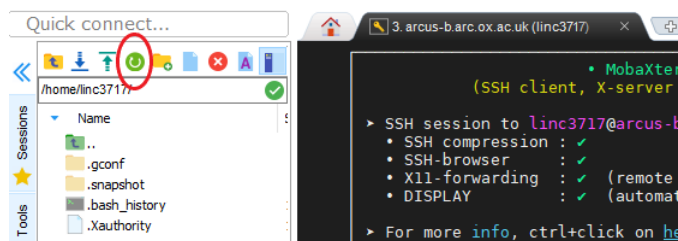


Figure 3: The Refresh button in MobaXterm's file browser.

We can now move `maine_coon.txt` into the folder `cats`, either by dragging-and-dropping in the graphical interface, or by:

```
mv maine_coon.txt cats
```

How do we check that the file was indeed moved? Just using `ls` now does not work, because `ls` only lists files in the current directory, which isn't `cats`. We need to change the current directory to `cats`:

```
cd cats
```

After doing this, your prompt should change from `~` to `~/cats`: this indicates that you have successfully *navigated* to the `cats` directory. Using `ls`, you can now verify that `maine_coon.txt` is in this directory. To return to the previous directory, use:

```
cd ..
```

The two dots `..` refer to the *parent directory* of your current location. In this case, that would simply be your home directory, since `cats` was created inside your home directory.

To delete a folder you can use `rmdir`. If you are in your home directory, you can try to remove `cats` with:

```
rmdir cats
```

but it will not work: the terminal will (helpfully) tell you that `cats` is not empty, so it refuses to delete it for you. We need to empty `cats` first before we can delete it. Navigate back inside there (`cd cats`) and delete the lone file there with:

```
rm maine_coon.txt
```

Return to the parent directory (`cd ..`) and try to remove the `cats` directory again (`rmdir cats`). This time it should work, and if you type `ls` you will find that it is completely empty, as expected. In terms of the contents of your home directory, we are now back where we started, but hopefully this short journey will have shed some light on how the terminal works.

Notice how we use an underscore instead of a space in our file name. Spaces in file names are not prohibited, but can potentially cause annoying problems which need to be carefully circumvented, so it is easier to avoid them altogether.

Instead of *moving* (`mv`) a file, you can also *copy* it using `cp`. The syntax is the same: `cp <file> <destination>`.

You can press `Tab` in the terminal to auto-complete file and folder names.

If you ever get hopelessly lost in the terminal, typing `cd` without any arguments will always bring you back to the safety of your home directory.

Unlike `rmdir`, the target of `rm` does not have to be empty for it to work. **BEWARE:** it is *extremely* difficult to recover a file once you have `rm'd` it. It does not go to a Recycle Bin; it is simply *gone*. So before running `rm` on something, make sure you really do want to delete it.

### Summary

- The prompt tells you useful stuff – including your current directory.
- `ls`: lists all files and folders in the current directory.
- `mkdir`: creates empty directories.
- `touch`: creates empty files.
- `mv`: moves files.
- `cd`: changes your current directory.
- `rmdir`: removes empty directories.
- `rm`: removes files.

## 3 Preparing an ORCA input file

ORCA is a computational chemistry programme developed by the group of Frank Neese. It takes an *input file*, containing all the instructions for the calculation, and (after a while) returns an *output file* containing all the results. ORCA can only be called using the terminal: there is no graphical interface, so you cannot (for example) draw a molecule in ChemDraw and click a button to run a calculation on it. In this section we will first analyse the structure of a simple input file, then look at how we can use molecular visualisation software such as Avogadro to help us generate input files.

"ORCA" isn't actually an acronym: it just came about because Frank Neese went on a whale-watching cruise in California.

A simple input file looks like the following:

#### h2o\_bad\_geometry.inp

```
! BP86 def2-SVP SP

*xyz 0 1
H -1.0 0.0 0.0
O 0.0 0.0 0.0
H 1.5 0.0 0.0
*
```

The first line

```
! BP86 def2-SVP SP
```

is a *keyword line*: it begins with a compulsory exclamation mark and contains several keywords which tell ORCA the level of theory as well as what type of calculation we want to do. In this case, **BP86** is a DFT functional, and **def2-SVP** a basis set. **SP** stands for *single-point*, which refers to the simplest kind of calculation where we only want ORCA to tell you the energy of a molecule.

The remainder of the input file tells ORCA what molecule you are interested in. In most computational chemistry software, molecules are not specified using connectivity information (i.e. what atom is bonded to what): instead, we specify the *geometry* of the molecule, or in other words, the 3D position of each atom in the molecule. We need three coordinates ( $x, y, z$ ) to describe the position of one atom, and that is exactly what the three lines

```
H -1.0 0.0 0.0
O 0.0 0.0 0.0
H 1.5 0.0 0.0
```

specify. For example, the first line tells ORCA that there is a hydrogen atom with the coordinates  $(-1, 0, 0)$  (the units are angstroms:  $1 \text{ \AA} = 10^{-10} \text{ m}$ ). Pictorially, our molecule could be drawn as in *Figure 4* below. Obviously, this is *not* a realistic geometry

Technically, the keyword **SP** is not needed: if you do not request a particular type of calculation, ORCA assumes you want a single-point. But it is a good idea to include it anyway, especially when starting out.

for water: it should not be a linear molecule, and the O–H bond distances should not be 1 and 1.5 Å! So, if we were to actually run this calculation, we should not be surprised to find that it is a rather unstable arrangement of atoms. When performing a single-point calculation, ORCA does not optimise the geometry of the molecule, and will just use whatever you give it.

It remains to explain the two lines surrounding the “coordinate block”. The asterisks are there to signal the start and the end of the coordinates respectively. `xyz` tells ORCA the format in which we are providing the coordinates, namely, as  $(x, y, z)$ -coordinates. `0` and `1` are the *charge* and *multiplicity* of the molecule respectively. Since water is a closed-shell molecule, its overall spin  $S$  is zero, and the multiplicity (defined by  $2S + 1$ ) is one (a singlet state).

For another example, if you had a radical anion species, you would change the line to `*xyz -1 2`.

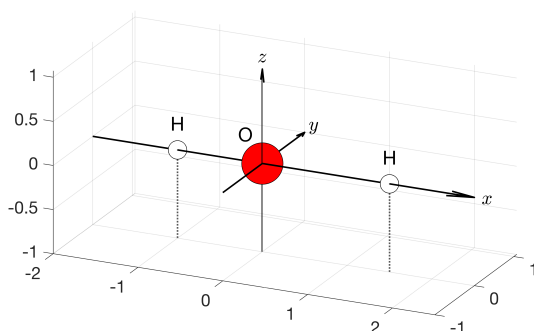


Figure 4: Our initial water geometry.

If we want to obtain a more realistic value for the energy of a water molecule, we need to first *optimise* the molecular geometry. That is to say, we need to instruct ORCA to find the minimum-energy geometry, and *then* calculate the energy of water at that minimum-energy geometry. This can be done by simply replacing the `SP` keyword with `Opt`, for optimisation. With this keyword, ORCA will take your initial geometry and try to tweak it a little bit at a time until it reaches a minimum energy.

Bear in mind that to get a sensible optimised geometry, the initial geometry that you give ORCA has to be at least somewhat reasonable. Our linear geometry is, unfortunately, not quite reasonable enough. But since we are dealing with water, the input file is relatively easy to construct by hand, and we could probably come up with a more sensible starting geometry without thinking too hard. What if, however, we wanted to look at larger, more interesting molecules? Enter Avogadro.

When you first open Avogadro, you will see a blank canvas on which you can “draw” a molecule. The drawing interface is fairly intuitive; feel free to play around with it a little. Once you’ve drawn something, head over to the **Auto Optimisation Tool** (Figure 5):

The better your initial geometry, the faster the optimisation will be, too.

You can rotate the camera angle by holding down the middle mouse button, pan by holding down the right mouse button, and right-click on atoms to delete them if you make a mistake.

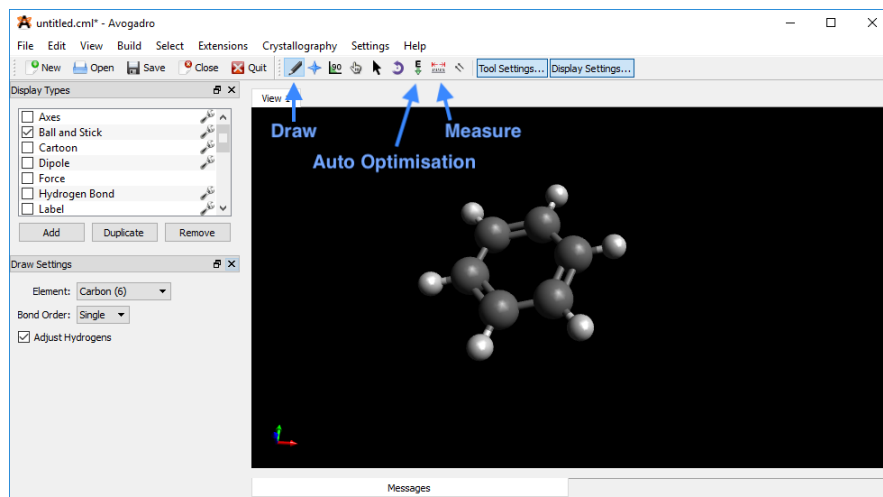


Figure 5: The various tools available in Avogadro.

Press the **Start** button, and Avogadro will perform a quick optimisation of your molecular geometry. Once the molecular geometry no longer changes, stop the optimisation and select (from the top) **Build > Cartesian Editor...** where you will find that Avogadro has prepared the entire coordinate block for us in the correct format! All that remains is to copy and paste the coordinate block into the ORCA input file and add the other lines detailing the level of theory and so on.

### Summary

- ORCA input files minimally consist of a *keyword line* and a *coordinate block*. The charge and multiplicity must also be specified.
- Coordinates are specified as  $(x, y, z)$ -coordinates in angstroms.
- You can use Avogadro to generate coordinate blocks for use with ORCA.

This optimisation is performed using a *force field*, which represents a very low level of theory: it is good enough to generate initial geometries for the high-level ORCA calculations, but it is not quite accurate enough to draw serious conclusions from (or to publish!).

## 4 Submitting jobs to the cluster

To run a calculation, we first need to get an input file onto the cluster. There are two main ways to do this. The first is to prepare the input file on your own computer, then drag-and-drop it into the MobaXterm pane. **Make sure you use a plain text editor, such as Notepad++, to prepare input files.** Programmes like Microsoft Word will not work as they add extra information about text formatting and layout.

There is a third way, which is to use terminal-based text editors such as vim, nano, or emacs. Although these are powerful tools, they also come with a steep initial learning curve.

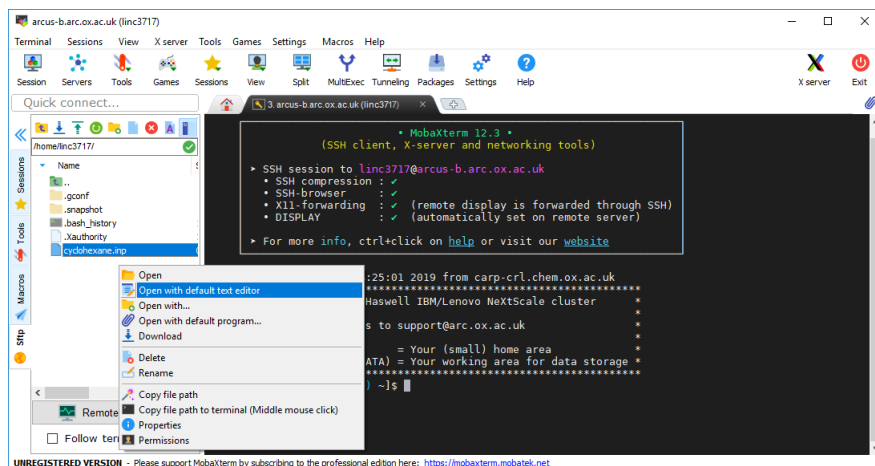


Figure 6: Accessing the default text editor in MobaXterm.

Alternatively, you can edit files on the cluster using the text editor built into MobaXterm. Once you have a file on the cluster (either by creating it from the terminal using `touch`, or by dragging-and-dropping), you can right-click it in the MobaXterm browser and select **Open with default text editor** (Figure 6).

Your first exercise will be to calculate the energy of cyclohexane, using the geometry pre-optimised by Avogadro. We will stick with the same level of theory as before, namely BP86 and def2-SVP, as this is capable of reaching “chemical accuracy” while also being relatively computationally cheap. Your input file will look somewhat like the snippet below. We have suggested the name `c6h12_sp.inp`, but you can give it any name you like, as long as you start with a letter and end with the file extension `.inp`.

`c6h12_sp.inp`

```
! BP86 def2-SVP SP

%pal nprocs 16 end

*xyz 0 1
... (insert coordinates from Avogadro here) ...
*
```

The new line

```
%pal nprocs 16 end
```

tells ORCA to use 16 CPU cores for the computation (which will speed it up significantly), and so we will use it henceforth.

To run a calculation, we need to submit it to the job scheduling software that manages the cluster. This requires a script, which you can obtain by running the following command in your terminal. Note that pasting text into the terminal (this particular one in MobaXterm, at least) is done using a right-click, not Ctrl-V.

```
source <(wget -O - - https://raw.githubusercontent.com/duartegroup/sbmcc/master/cc20.sh)
```

This will give you a new file called `qorca` in your home directory. It should be noted that `qorca` is not actually the ORCA programme itself; it merely requests the cluster to run the ORCA job using your input file. To submit the input file you prepared earlier, first `cd` to the directory containing it, then run:

```
qorca c6h12_sp.inp
```

You should see a confirmatory message that your job has been submitted. To check the progress of your job, you can use either:

```
squeue -u <username>
```

which will print a list of your running jobs with their status (`PD` = pending and `R` = currently running). Once a job is complete, it will no longer show up. Alternatively, you can track the progress of the job using:

```
tail c6h12_sp.out
```

which prints the last few lines of the output file. Notice that we are using the `.out` extension here (which corresponds to the output file) instead of our original `.inp`. When it finishes a calculation, the output file will have something similar to the following, which you can observe using `tail`:

All computations on the `arcus-b` cluster are automatically allocated at least 16 CPUs, so it would be a waste to not make full use of it!

You *must* do this in the terminal. MobaXterm’s graphical file browser is not capable of submitting jobs.

If the job ends (no longer shows up when using `squeue`) but the file does not end with this line, please consult a demonstrator.



```
****ORCA TERMINATED NORMALLY****  
TOTAL RUN TIME: 0 days 0 hours 0 minutes 16 seconds 827 msec
```

### Summary

- Input files can be prepared on your own desktop (using e.g. Notepad++) or on the terminal (using MobaXterm's editor).
- You can submit jobs on the cluster from the terminal using the `qorca` script.
- Job progress can be monitored using either `squeue` or `tail`.

## 5 Comparing cyclohexane geometries

If your job ran successfully, the output file should contain in excess of 1000 lines. For a more complicated calculation, such as an optimisation (which you will do soon), this number can grow very rapidly! How do we pick out the important information from all of that text?

One way is to manually inspect the file and look for certain keywords, which we will do now. Open up the output file `c6h12_sp.out`, either by dragging it back to your desktop and using Notepad++, or by using the built-in MobaXterm editor. Look for a line near the end that reads **FINAL SINGLE POINT ENERGY**: this is the energy of your molecule in that particular geometry. Take note of this number.

Alternatively, we can use software to help us parse the output file. Avogadro is capable of doing this (to some extent). If you haven't already done so, copy your output file back to your own computer, then open it with Avogadro. It will display the final geometry from the ORCA calculation. Since this was a single-point calculation, ORCA does not change the geometry, so it will be exactly the same as the geometry you submitted. Use the **Measure Tool** (Figure 5) to measure some of the key properties listed below:

- C–C and C–H bond lengths (for both equatorial and axial protons)
- C–C–C and H–C–H bond angles

Once you have recorded these somewhere, make another copy of your input file, but change the **SP** keyword to **opt**. This tells ORCA that we want to optimise the geometry. Submit this job (the calculation itself should not take more than 2 minutes) and analyse the output using the same steps as before. This time, when you open the output file, Avogadro will display the geometry as optimised by ORCA, which will be different from the previous geometry. Compare the energies, bond lengths, and bond angles obtained from the two calculations. How different are they?

### Summary

- Single-point calculations involve ORCA calculating the energy of one single given geometry, whereas optimisations allow ORCA to search for a minimum-energy geometry.
- Simple pieces of information such as energies can be obtained by visual inspection of ORCA output files.
- Avogadro is useful for visualising final geometries.

We already learnt that bond lengths are specified in angstroms. You should now do some research to find out what the units of energy in computational chemistry are, and how they relate to familiar units such as kcal/mol or kJ/mol.

## 6 Conformational analysis

In this next exercise we will study an  $\alpha$ -halocarbonyl compound. Our prototype molecule is (*S*)-2-fluoropropanal, for which the starting geometry can be downloaded from <https://www.echemsimulator.com/2018/05/20/2-fluoropropanal-geometry/>:



`//raw.githubusercontent.com/duartegroup/sbmcc/master/s2fp.xyz`. Open up the `.xyz` file with a text editor. You should find that it contains the desired coordinate block, along with two more lines at the top. The first line (**10**) is the number of atoms in the molecule, and the second line is a comment line which can be anything. This is the standard format for `.xyz` files.

We will first look at how the energy of the molecule varies with the O–C–C–F dihedral angle. To do this, we need to do a series of optimisations where the dihedral angle is constrained to a particular value. This is called a *potential energy surface (PES) scan*, and can be performed by invoking an additional `Scan` line. For this exercise, we will use Hartree–Fock theory instead of DFT to speed up the calculations, so we replace the original BP86 def2-SVP with HF-3c.

If we did not constrain the dihedral angle, it would simply be optimised to a (local) minimum, and we would not learn anything about the rest of the PES.

#### s2fp\_scan.inp

```
! HF-3c Opt

%pal nprocs 16 end

%geom Scan <SCAN_COORDINATES> end end

*xyz 0 1
... (paste the coordinates from the .xyz file in here) ...
*
```

We still need to specify the coordinate that we are scanning over. In this case, it is a dihedral angle, and in ORCA this is specified using the syntax

`D <atom1> <atom2> <atom3> <atom4> = <initial>, <final>, <number_of_steps>`

We can set the initial value to 60, the final value to 420, and the number of steps to 19. Now we just need to specify which atoms form the dihedral angle of interest. In ORCA this is done using the *atom number*: the first atom listed in the coordinate block has number 0, the second atom has number 1, and so on. The fastest way of finding the atom numbers is to use software such as Avogadro.

420° is the same as 60°, of course, and using 19 steps means we get precisely a 20° difference between each step. Why start at 60° instead of 0°? It turns out that our initial geometry has this dihedral angle close to 60°, and it's always better to start a scan from a value that is close to the initial geometry.

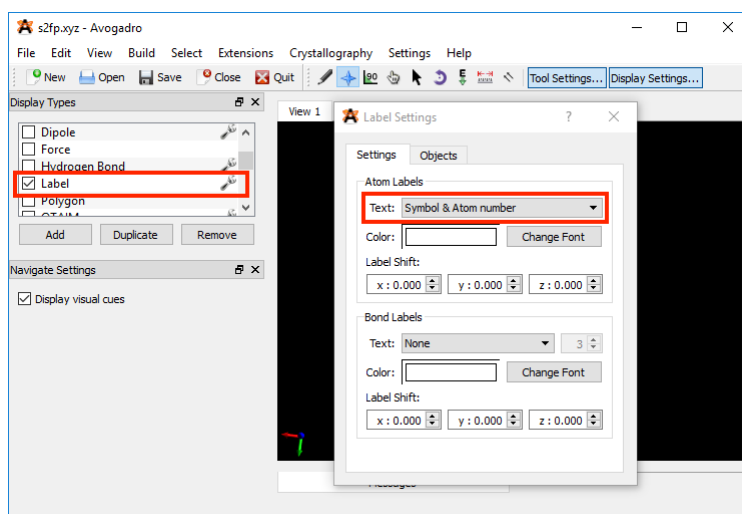


Figure 7: Enabling atom labels in Avogadro.

Open up the `.xyz` file you downloaded earlier with Avogadro. In the left-hand **Display Types** pane, scroll down to **Label** and make sure the checkbox next to it is ticked

(Figure 7). Then, click the spanner on the right. This will bring up a settings pane. Under **Atom Labels > Text**, select **Symbol & Atom number**. Close the settings pane and you should now see that each atom in the molecule is labelled with the element symbol as well as the atom number.

**There is a catch: atom numbers in Avogadro start from 1, whereas ORCA starts from 0.** So, whenever you are preparing an ORCA input file, always remember to subtract 1 from the number Avogadro shows. Find the atom numbers that we are interested in: these are respectively the carbonyl oxygen, carbonyl carbon,  $\alpha$ -carbon, and fluorine. Substitute these into the input file to get:

#### s2fp\_scan.inp

```
! HF-3c Opt

%pal nprocs 16 end

%geom Scan D 8 0 1 6 = 60, 420, 19 end end

*xyz 0 1
... (paste the coordinates from the .xyz file in here) ...
*
```

Submit the job on the cluster as before. This calculation should take just under 10 minutes. When it is complete, open up the .out output file and search for the text

The Calculated Surface using the 'Actual Energy'

Under this line there are two columns of numbers: unsurprisingly, these are respectively the value of the dihedral angle, and the energy. Use a programme of choice to plot a graph of energy against the dihedral angle (it is a good idea to first convert the energies to more understandable units). You should find that there are two minima in the resulting graph, which represent the two conformers. Answer the following questions:

- What are the relative populations of the two conformers at 298 K? Use the Boltzmann distribution:

$$\frac{p_i}{p_j} = \exp\left(-\frac{\varepsilon_i - \varepsilon_j}{k_B T}\right)$$

- What is the activation barrier for the conversion of the low-energy conformer to the high-energy conformer? What is the corresponding rate at 298 K? Use the Eyring equation (assume the transmission coefficient  $\kappa = 1$ ):

$$k = \frac{\kappa k_B T}{h} \exp\left(-\frac{\Delta E}{k_B T}\right)$$

- Qualitatively, how might you expect your results to change if we included solvation by a polar solvent such as DMSO? (If you have spare time and would like to run this calculation, consult a demonstrator.)

If you wish to inspect the optimised geometries, ORCA also outputs .xyz files for each step of the scan. For example, <filename>.001.xyz corresponds to the first step of the scan (which has the dihedral angle set to 60°), and so on.

If the **Display Types** pane is not visible, click on **Display Settings...** at the top-right corner to open it.

The order in which you specify the numbers matters! Any other combination will specify a *different* dihedral angle from what we want. You can reverse them (D 6 1 0 8 = ...), but the corresponding dihedral angle will have the opposite sign, so you should start the scan from -60°.

There is also a surface using the "SCF energy", but this neglects other contributions such as dispersion.

In principle, these should be calculated using Gibbs free energies, not pure electronic energies as we are doing here.

### Summary

- .xyz files consist of two lines, plus the coordinate block used for ORCA input files.
- Atom labels start from 0 in ORCA, but start from 1 in Avogadro.
- In a PES scan, ORCA calculates how the energy of a system changes as one or more parameters are varied.

## 7 Finding a transition state

In the previous section we introduced the concept of a PES scan. By judicious choice of the parameter(s) being scanned over, it is possible to use this to study the course of a chemical reaction. The *reaction coordinate* of a simple reaction can be characterised by one single parameter, such as a bond length or a bond angle. Here we will study the intramolecular tautomerisation of malonaldehyde (*Figure 8*):

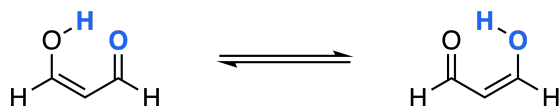


Figure 8: Intramolecular tautomerisation of malonaldehyde.

As the tautomerisation proceeds (from reactant to product), the distance between the highlighted oxygen and hydrogen atoms smoothly decreases. So, we can perform a bond length scan. The syntax is similar to the dihedral angle scan we did previously. You can obtain the initial geometry (already pre-optimised with BP86/def2-SVP) from <https://raw.githubusercontent.com/duartegroup/sbmcc/master/malon.xyz>.

Our starting “bond length” (it isn’t really a bond yet) is the internuclear separation in the initial geometry, which should be  $\sim 1.48$  Å. The final bond length can be determined from an optimised geometry of the product – which, in this case, is exactly the same as the reactant – so we can measure the final bond length as being  $\sim 1.06$  Å. We will use 12 steps, and BP86/def2-SVP as the level of theory. Your input file should look like:

### malon\_scan.inp

```
! BP86 def2-SVP Opt

%pal nprocs 16 end

%geom Scan B 8 4 = 1.48, 1.06, 12 end end

*xyz 0 1
... (paste coordinates here) ...
*
```

(Notice we use **B** for bond length, instead of **D**, inside the **Scan** line.) Submit this input file: it should take around 5 minutes to run. Once it is complete, plot the potential energy surface in the same way as before.

You should find that somewhere in the middle, there is a maximum in the energy. Find the .xyz file which corresponds to this maximum (don’t forget that our scan was in decreasing order of bond length, so .001.xyz corresponds to the largest distance). This represents a geometry that is very close to the transition state. We will now use that geometry as a starting guess for a transition state search. Before proceeding, make a note of the key O–H distances in this structure.

It is not *exactly* the transition state, because the surface is not smooth: we only used discrete values of the bond length.

For a transition state search, there are several changes to make to the input file:

- The `OptTS` keyword requests an optimisation to a transition state (instead of `Opt`, which optimises to a minimum);
- We also add the `Freq` keyword which requests vibrational frequencies. This is very important for characterising transition states, which should have one (and only one) imaginary frequency;
- Since we are not doing a scan any more, we can remove the `Scan` line;
- In place of it, we add a line (`Calc_Hess`) which instructs ORCA to calculate a Hessian before beginning the optimisation: this helps to guide the optimisation and makes it much more likely to succeed.

All in all, this gives us:

malon\_ts.inp

```
! BP86 def2-SVP OptTS Freq

%pal nprocs 16 end

%geom Calc_Hess true end

*xyz 0 1
... (paste coordinates here from .00X.xyz file) ...
*
```

This calculation should take under a minute. Open up the optimised transition state geometry with Avogadro and compare the O–H bond lengths with the bond lengths you noted down previously. How different are they? Which geometry is more symmetric, and is that in line with what you expect based on the symmetry of the system under investigation?

As the final part of this tutorial, we will look at the vibrational modes of the transition state you have found. Search for the section containing **VIBRATIONAL FREQUENCIES**. You should find that the first six “vibrational” frequencies are zero. These are followed by one *negative* frequency, which is (unfortunately) the conventional way of representing *imaginary* frequencies. The rest of the frequencies should be positive.

We need to visualise this imaginary mode to make sure that it is indeed the vibrational mode that we are interested in – namely the reaction coordinate. To do this, we use an ORCA helper utility called `orca_pltvib`: this is capable of reading the `.hess` file which should have been produced by the TS calculation. Run the following two commands:

```
module load ORCA/5.0.1-gompi-2019b-dba
orca_pltvib malon_ts.hess 6
```

(Don’t use the `.001.hess` file – that is the *initial* Hessian which we told ORCA to calculate using `Calc_Hess`.) The number 6 refers to the imaginary frequency (remember ORCA counts from 0, so the numbers 0 through 5 correspond to the zero-frequency modes). Once you have done this, there should be a new `.hess.v006.xyz` file in the directory. Open it in Avogadro and select **Extensions > Animation...** We recommend ticking both **Dynamic Bonds** and **Loop**. When you press the Play button, you should see that the imaginary mode does indeed correspond to the reaction coordinate, which brings reactant to product via the transition state we found.

The imaginary frequency is related to the fact that a transition state is a maximum along only one direction, which is the reaction coordinate. Maxima are associated with a negative force constant, and the corresponding frequency is proportional to the square root of the force constant.

The zero-frequency “vibrations” are actually translations and rotations.

If you’re wondering what a Hessian is: it’s a matrix containing all the second derivatives of energy with respect to nuclear positions.

**Summary**

- The maximum in a PES scan can be used as a starting geometry for a transition state search.
- Transition states are characterised by one imaginary frequency.
- Vibrational modes can be visualised using `orca_pltvib`.

## 8 Other resources

### Books

- J. Harvey, *Computational Chemistry*. Oxford Chemistry Primers, 2018. ISBN: 9780198755500.
- S. Bachrach, *Computational Organic Chemistry (2<sup>nd</sup> Edition)*. John Wiley & Sons, 2004. ISBN 9781118291924.
- C. J. Cramer, *Essentials of Computational Chemistry (2<sup>nd</sup> Edition)*. John Wiley & Sons, 2004. ISBN: 978-0-470-09182-1.
- F. Jensen *Introduction to Computational Chemistry (2/3<sup>rd</sup> Edition)*. John Wiley & Sons. ISBN: 978-1-118-82599-0.
- H. P. Hratchian and H. B. Schlegel, Chapter 10 - Finding minima, transition states, and following reaction pathways on *ab initio* potential energy surfaces (pg 195-249) in *Theory and Applications of Computational Chemistry*. link: <http://www.pitt.edu/~jordan/chem3430/289.TACC.pdf>

### Websites

- Anna Tomberg – <https://www.youtube.com/user/IaNuisha> (search for Avogadro / ORCA videos)
- ORCA manual and forums – <https://orcaforum.kofo.mpg.de/> (need to register first)
- ORCA Input Library – <https://sites.google.com/site/orcainputlibrary/home>