Introduction to Computational Chemistry
Exercises Part 1

Eno Paenurk & Patrick Finkelstein & Felix Pultar

April 26, 2023

Goals:

- Accessing the cluster
- Using terminal/basic bash commands
- Recognizing and resolving general errors

The use of the terminal may be non-intuitive if you have never used it before. For this reason, throughout the exercises there will be tasks that can be accomplished in two ways, depending on your preference of terminal commands or more standard editors.

Instructions in these boxes are for those who prefer the more intuitive way, with external editors, and clicking at stuff instead of using bash commands.

If you prefer to work with the terminal/bash commands, you will find the instructions in these dashed boxes. You can consult the Handout and Section 9 here for extra help.

If you have followed our ICC-Workshop-preparations guide you can move on directly to Section 3. If not, Section 1 is relevant for users from ETH connecting to Euler, while Section 2 is for users from Yale with access to Grace.

1 Euler

1.1 Accessing Euler

Windows

- Open MobaXterm
- Select Sessions - New Session
- Select SSH and insert the remote host (euler.ethz.ch). Tick the "specify username" box and type your username. Then click OK.
- Type your ETH password (note that no characters will be shown as you type the password)

You can later skip this step and just select euler.ethz.ch from "Recent sessions" when you open MobaXterm.

Mac/Linux

Open your terminal and type `ssh username@euler.ethz.ch` and then enter your password.

1.2 Accessing files from Euler

Windows

- Open MobaXterm
- Select Sessions - New Session
- Select SFTP and insert the remote host (euler.ethz.ch). Type your username. Then click OK.
- Type your ETH password (note that no characters will be shown as you type the password)

You can again later skip this step and just select euler.ethz.ch from "Recent sessions" when you open MobaXterm (it will have a different icon to distinguish it from the SSH connection).

Mac/Linux

- Open FileZilla
- Type in the Host (euler.ethz.ch).
- Type your username and password, and the port (22).
- Click on Quickconnect

2 Grace

2.1 Accessing Grace

Follow the instructions at `https://docs.ycrc.yale.edu/clusters-at-yale/access/ssh/` to access the computer cluster. The general instructions are similar to those in Section 1.2, but you will first need to generate your SSH key.

2.2 Accessing files from Grace

You can proceed similarly to Section 1.2, but replace euler.ethz.ch with transfer-grace.hpc.yale.edu. Note that you should do file transfers on the transfer node (i.e., not just the grace.hpc.yale.edu host). You can find more information at `https://docs.ycrc.yale.edu/data/transfer/`. If you use FileZilla, you will need to first add your private SSH key (see Section 2.1) with Edit-Settings-Connection-SFTP-Add key file.

3 General Orientation

In the beginning it is common to "get lost" in some random folder. To find out where you are, you can type

```
pwd
```

which stands for "print working directory". The output could be something like this.

```
/cluster/home/username/folderX/folderY/001
```

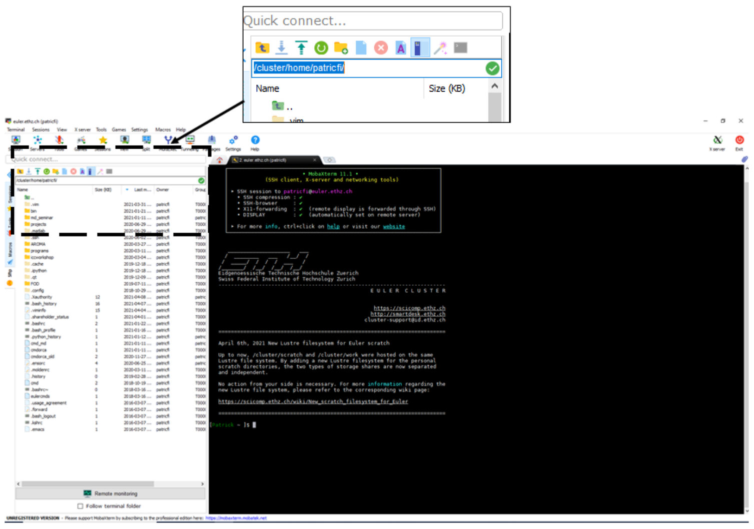
To view the files in this directory, you can either type

```
ll
```

or

```
ls
```

or you can use the file-explorer in the left sidebar of MobaXterm (or in the SFTP client of your choice). Either click your way through the folders, or use the command to print your current directory and copy it to the directory bar of the explorer.



```
                    • MobaXterm 11.1 •
          (SSH client, X-server and networking tools)

 ➤ SSH session to patricfi@euler.ethz.ch
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Eidgenoessische Technische Hochschule Zuerich
Swiss Federal Institute of Technology Zurich
-------------------------------------------------------------------------
                                                      E U L E R  C L U S T E R

                                                      https://scicomp.ethz.ch
                                                     http://smartdesk.ethz.ch
                                                   cluster-support@id.ethz.ch

=========================================================================

April 6th, 2021 New Lustre filesystem for Euler scratch

Up to now, /cluster/scratch and /cluster/work were hosted on the same
Lustre file system. By adding a new Lustre filesystem for the personal
scratch directories, the two types of storage shares are now separated
and independent.

No action from your side is necessary. For more information regarding the
new Lustre file system, please refer to the corresponding wiki page:

https://scicomp.ethz.ch/wiki/New_scratch_filesystem_for_Euler

=========================================================================
[Patrick ~ ]$ 
```

In most of cases pressing `Ctrl-C` within a terminal environment will simply send an interrupt or terminate signal to whatever process or application is running. For most terminals, to copy anything from inside the terminal, it suffices to select/highlight the object to be copied, click on the explorer bar, and then press `Ctrl-V`. If you plan to paste something into a terminal environment simply right-click or middle-click (depending on the setup) into the terminal. For the default Mac terminal, you should still use the select + `Cmd-C` and `Cmd-V` combination to copy and paste.

4 Folder Setup

We recommend to not use spaces in any folder or file names. Instead, use the underscore or hyphen to separate parts of a name. Make a folder for this workshop called “icc-2023”.

Within the file explorer, right-click and select “New directory” (MobaXterm) or “Create directory” (FileZilla) and enter a suitable name.

You can use the `mkdir` command to make a new directory. Type

```
mkdir dummy
```

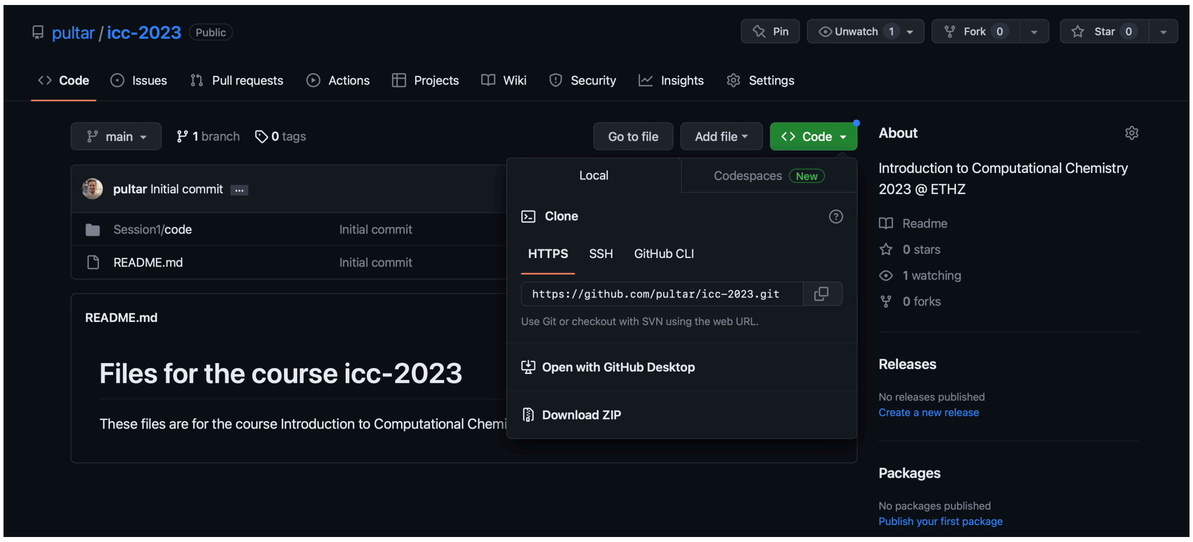
to create a directory with the name “dummy”.

Within the "icc-2023" (remember to change into the folder first), create a subfolder called "workspace" and within this folder, make another folder for each of the sessions, i.e. part1, part2 etc. Display all the subdirectories within the "icc-2023" directory to verify you have created the correct hierarchy.

5 Download the Lecture Material

For the exercise today and coming sessions, we will provide you with files that can be downloaded from GitHub, an online hosting service: `https://github.com/pultar/icc-2023`. GitHub allows seamless updates of the lecture material (e.g. in the case we find a mistake) at a single repository. This workflow is advantageous over, e.g. many e-mail attachments.

First make a new folder within the "icc-2023" directory called "materials". Next, visit the URL (see above) in your browser, click the big green button `Code`, and hit `Download ZIP`. You will get a zip archive that you can extract and transfer to the materials folder you just created. This workflow is easy as it does not use the command line but you will have to come back to the website and repeat the steps every time you want to update the lecture material.



```
https://github.com/pultar/icc-2023.git
```

Use Git or checkout with SVN using the web URL.

Files for the course icc-2023

These files are for the course Introduction to Computational Chemi

Introduction to Computational Chemistry 2023 @ ETHZ

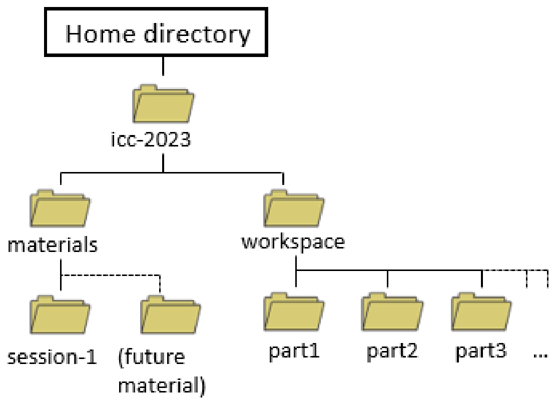
Change to the "icc-2023" folder. The following command will download the latest version of the material and put it into a new folder, aptly named "materials":

```
git clone https://github.com/pultar/icc-2023.git materials
```

These commands will update the material in the future:

```
cd icc-2023
git pull
```

Your folder hierachy should now look like this:



6 Bash Commands and Navigation

You will find a file `ex1_1.sh` within the "code" folder you downloaded from GitHub (most likely in /icc-2023/materials/session-1/code). Copy this file into /icc-2023/workspace/part1 by making use of the `cp` command. Within the target folder type

```
cp ../../materials/session-1/code/ex1_1.sh .
```

For most of this you can simply use the tab key to auto-complete the path.
The general idea here is to have all the files we provide in the documents folder and to work on the scripts etc. in the workspace subfolders (by copying the relevant files).
Have a look at the contents of the file. You can either do this directly in the terminal or by using an external editor.

Go to the location of the file in the explorer window (either by clicking on the folders or by typing `pwd` and copying the path). Right-click on the file and select "Open with default editor" in MobaXterm or View/Edit in FileZilla.

You can display the contents of the file by either using the command `cat` or using the text editor `vim`. As `cat` prints the whole contents of the file, we recommend use it only for short files. To use it, simply type

```
cat ex1_1.sh
```

Alternatively, you can also open the file in vim

```
vi ex1_1.sh
```

To exit vim just type

```
:q
```

and hit enter.

The file contains a short script that will echo your username. Try to execute the file by typing

```
./ex1_1.sh
```

You will notice that you do not have the permission to execute this file. Every file has three types of permission: reading, writing, and executing. The default is to only have permission for reading and writing, so you will have to add the third one.

Right-click on the file in the explorer bar and select Permissions (MobaXterm) or File Permissions (FileZilla). Tick the User - execute box and click Apply (MobaXterm) or OK (FileZilla).

Linux file permissions are managed for three different user classes: owner, group, and public. Each of these can receive different permissions to either read, write, execute, or any combination thereof. The full set of permissions is given as an octal code consisting of three digits, one for each user class (first the owner, then group, and finally public). Each user class can be assigned either a permission (denoted as a 1) or no permission (denoted as a 0) for reading (r), writing (w), and executing (x). The `chmod` command to change permissions is derived by simply converting the binary representation given as rwx into the corresponding octal digit for all three user classes. An overview is shown below:

Octal Digit	rwx	Permission
4	100	Read only
5	101	Read and execute
6	110	Read and write
7	111	Read, write, and execute

Thus, to give yourself all permissions, while allowing the two other user classes (group and public) to only read your file, type

```
chmod 744 ex1_1.sh
```

Try again to execute the file. You should now get a new error, which is because we provided you with a file written in a Windows text editor. As mentioned in the lecture, these are incompatible with Linux and a frequent cause for errors. There is, however, an easy way to convert the file using a bash command.

```
dos2unix ex1_1.sh
```

Notice how you can use the tab button to complete bash commands. For instance, if you type “do” and press tab twice you will be shown something like

```
do  domainname   done  dos2unix     dotlockfile  doxygen
```

which are all bash commands starting with “do”. If you do the same after only typing “d” you will be asked if you want to display all options—in this case it would not be very useful as there are over a 100. Just press “n” to opt out from displaying all of them (or “y” if you do want to know all the options).
With the converted file, try again to execute it.

7 Loading Modules

In the following weeks we will require a number of modules (applications). To get an overview of the modules that you currently have loaded in your profile, type

```
module list
```

You can add different modules that allow you to use resources like MATLAB, Java, RStudio, etc. A full list can be found on: `https://scicomp.ethz.ch/wiki/Euler_applications_and_libraries`. To see exactly which modules you can load, type

```
module avail
```

To have access to the new module system, type

```
env2lmod
```

If you would now want to use the newest MATLAB, for example, you should type

```
module load matlab/R2022a
```

which would enable you to call MATLAB on Euler.
Removing one or all modules is also simple. To remove a specific module (let’s remove the MATLAB module we just added), type

```
module unload matlab/R2022a
```

To remove all the modules at ones, you can use

```
module purge
```

Execute this last command and check that you module list is indeed empty by using the `module list` command.

In the coming weeks we will be using a number of key modules on a regular basis, so it would be practical to have them loaded automatically each time you login to Euler. We will load two of these today. For this, you need to modify a hidden file in your home directory. Hidden files are not shown if you type

```
ls
```

FileZilla displays hidden files by default, but MobaXterm does not (hidden files start with a ".")

In MobaXterm, you can show hidden files in the explorer by clicking on the third icon from the right (Show hidden (dot) files). Open the .bash_profile file in your home directory. At the bottom of the file start a new line and add:

```
env2lmod
module load gcc/9.3.0 openmpi/4.1.4 orca/5.0.3
```

Click on File - save and confirm when the prompt shows up to ask you if you are sure. In FileZilla, you need to additionally close the file and agree to uploading the file back to the server.

The `ls` command without any options lists the contents of the directory but ignores all entries that start with a "." This means that we need to add an option that will reveal these entries too: the `-a` flag.

```
ls -a
```

You should now see a larger number of files than before. Open the one we are interested in, the `.bash_profile` file, by typing

```
vi .bash_profile
```

There are two main modes within vim: insert and command mode. Pressing `i` will switch to insert mode, which you can verify by starting to type or by the word "insert" on the bottom left of the screen.
Navigate to the end of the file with your arrow keys, start a new line with enter, and type (or copy)

```
env2lmod
module load gcc/9.3.0 openmpi/4.1.4 orca/5.0.3
```

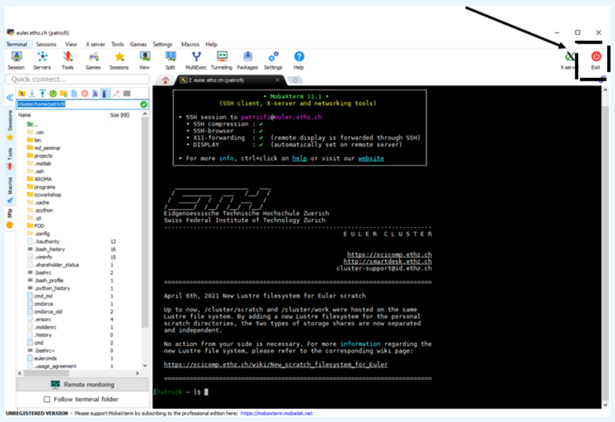
Press `Esc` on your keyboard to switch to command mode and type

```
:wq
```

You should see this typed out at the bottom left of your terminal screen. Press enter—this will write/save the changes (w) and quit (q).

Have another look at your module list (with `module list`). As you can see, the modules you added aren't there yet. They will appear once you source your `.bash_profile` file. There are two easy ways to do this.

The `.bash_profile` file is automatically sourced when you connect to Euler. This means you simply have to exit and reconnect. You can exit by clicking on the red button in the top right corner.


```
                    • MobaXterm 11.1 •
            (SSH client, X-server and networking tools)

 ➤ SSH session to patricfi@euler.ethz.ch
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Eidgenoessische Technische Hochschule Zuerich
Swiss Federal Institute of Technology Zurich
-------------------------------------------------------------------------
                                                    E U L E R  C L U S T E R

                                                     https://scicomp.ethz.ch
                                                    http://smartdesk.ethz.ch
                                                  cluster-support@id.ethz.ch

=========================================================================

April 6th, 2021 New Lustre filesystem for Euler scratch

Up to now, /cluster/scratch and /cluster/work were hosted on the same
Lustre file system. By adding a new Lustre filesystem for the personal
scratch directories, the two types of storage shares are now separated
and independent.

No action from your side is necessary. For more information regarding the
new Lustre file system, please refer to the corresponding wiki page:

https://scicomp.ethz.ch/wiki/New_scratch_filesystem_for_Euler

=========================================================================

[Patrick ~ ]$
```

Method 1
You can quickly exit and restart by typing

```
exit
```

and then pressing `R`.

Method 2
You can also directly source the file by typing

```
source .bash_profile
```

Make sure you are in your home directory when you do this.

If you now look at your module list, you should see the new modules added.

8 Submit your first job

In the next weeks, we will run a large number of CPU-intense jobs that are not suited to be run on the login node. Instead, these are best submitted to the cluster compute nodes. The following script will serve as an example.
Enter your part1 folder and copy the `ex1_2.py` file which we provided (follow the same steps as described in Section 6). Submit the script to the cluster with

```
sbatch --wrap="./ex1_2.py > ex1_2.out"
```

and monitor its progress with

```
squeue
```

There are two main (visible) stages for a job: pending (PD) and running (R). Once the job has finished, it will no longer be visible when you type `squeue`.
When the job has finished (this should be very fast once it has started running), check the ".out" and the slurm file that corresponds to your submitted job (for the latter, it will be labeled by the job ID number). Open the files with whichever method you prefer and study the contents:

- Did the job run correctly?
- If no, what could be the reason?
- How would you fix it?

The answer is given at the end of the exercise—but try to figure it out first!
Do what is required to make the job work (you should by now know all the necessary commands/procedures) and resubmit it.

9 Extra: vim

Here we provide an optional exercise, for those who are curious about vim and might want to use it more in the future.
Navigate to the ex1 folder of part1. You can open a file in vim by simply typing

```
vi dummy
```

where dummy would be the name of the file. If no file with this name exists, vim will open a new file with this name.
Navigating within vim is accomplished by using the arrow keys to move the cursor around. The mouse cursor is usually only useful for selecting (which means copying) content. A short selection of shortcuts that are useful is given below.

Keys	Description
`shift + g`	end of the file
`gg`	beginning of the file
`Home`	beginning of the line
`End`	end of the line

There are two main modes within vim: insert (by typing `i`) and command mode (default when you open the file, or accessed from insert mode by pressing `Esc`). Insert mode is used to write in the file, command mode is used for various commands, for example, to save, undo, delete lines, or exit the editor. Within command mode, every command starts with a colon ":" (*e.g.,* `:q` to quit).

vim command	Description
`w`	saves all changes
`q`	quits the editor
`q!`	force quit without saving (useful for undoing many changes)
`wq` *or* `x`	save and quit
`u`	undo changes
`/dummy`	search for the dummy, go to next occurence with "n", go to previous with "N"

Practise by making a new file ex1_3.sh in the ex1 folder with the following content:

```
#!/bin/bash
echo "Mom, look, I wrote a script!"
```

The first line might look strange, but it is required for the script to run. Now, try to run it either directly in the terminal or by submitting it to Euler.

Solution to the job error for ex1_2.py

There are two issues with the submission.

- If you want to submit a script directly, you will need to modify the permissions for this. You can circumvent this by instead calling python on the file:

  ```
  sbatch --wrap="python ex1_2.py > ex1_2.out"
  ```

- As just mentioned in the previous issue, the input file is a python script, so you will need a python module. Simply add the latest available python version on the cluster to your module list. (Make sure you added the gcc module as instructed in Section 5)