High Performance Computing
Exercise Sheet 1

HS 25
Dr. Douglas Potter

https://www.astro.uzh.ch/en.html
Teaching Assistants:
Cesare Cozza, cesare.cozza@uzh.ch
Mark Eberlein, mark.eberlein@uzh.ch

Issued: 19.09.2025
Due: 26.09.2025

Exercise 0 [Course workspace & hand-in]

All documentation for this course (slides, exercises, etc.) can be found within the *Microsoft Teams* workspace. You should have received the invitation from Dr. Potter.

Concerning the exercise sessions, we ask you to hand-in some of the solutions to the problems (the ones labeled by ***Hand-in:***). Don't worry if you couldn't solve the exercise: in that case describe your attempt and what went wrong. Once a new exercise sheet is released, you will find it as an assignment within the *Teams* workspace. For this week, we ask you to attach a file with your solutions into the course workspace (you can either upload your file or create it directly within *Teams*).

Exercise 1 [Setting up]

- Windows 10, 11: More detailed tutorial and information can be found here: https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-11-with-gui-support#1-overview. You need to get bash on windows:
  1) Open Settings, go to Update & Security, For developers: turn on "Developer Mode"
  2) Open Control Panel, go to Programs and Features, Turn Windows Features On or Off, and turn on "Windows Subsystem for Linux". Press OK and reboot.
  3) Go to the Windows store and install the "Ubuntu" app.
  4) Launch the app, choose a username and password (and remember these :)).
- Windows 11: X11 forwarding has become more convenient using `WSL2`, so everything should work well (you might need to update `WSL` from your command prompt: `wsl --update`)
- Windows 10: You also need an X server:

Download `VcXsrv` from http://sourceforge.net/projects/vcxsrv/ and install. Run `VcXsrv`, go back to your Ubuntu app and type `export DISPLAY=localhost:0`. Make sure to start `VcXsrv` before connecting to ela.cscs.ch.

- Older Windows versions: download and install `Putty`, `pageant` and `VcXsrv`. Download `Putty` from http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html and install. Use the following configuration:
  To add a session use:

  Session:
  Host name: ela.cscs.ch
  Connection, Data:
  - Auto-login username: stud?? (enter username)
  SSH, Auth:
  - Check "Allow agent forwarding"
  SSH, X11:
  - Check "Enable X11 forwarding"
  - Add localhost "localhost:0"

  (Optionally change colours)
  Window, Colours:
  - Default Foreground: 0,0,0
  - Default Background: 255,255,255
  - Default Bold Background: 165,5,37

  Go back to Session and enter a name "Saved Sessions" and click "Save".

  Download `VcXsrv` from http://sourceforge.net/projects/vcxsrv/ and install. **Make sure you started `VcXsrv` before trying to connect to ela.cscs.ch**

- Mac OS X: If you don't have XQuartz installed, go to http://xquartz.macosforge.org/ and download it. Open the package to install. Click: Continue, Continue, Continue, Install. Then enter the superuser/admin password and click Install Software.

- Linux: you don't need to do anything!

Exercise 2 [Logging in to Eiger]

Open a terminal and log in to Eiger using the login name and password which you have been provided with. In general, logging in to a remote system can be done with `ssh <login name>@<host>`. To log in to Eiger you have to first log in to the frontend `ela.cscs.ch`. From there you can connect to Eiger using `ssh eiger` and the same password. In order to get screen forwarding, use the `-Y` or `-X` flag (e.g. `ssh -X course01@ela.cscs.ch`).

NOTE: CSCS has recently added a Multi Factor Authentication (MFA) Protocol to log in to eiger. You can find the instruction on how to set it manually via web based service or via SSHService through command line at https://docs.cscs.ch/access/mfa/ . After the MFA authentication is set you can connect with SSH and set authomaticallt the SSH key following the steps at https://docs.cscs.ch/access/ssh/.

Now log back out from Eiger using either `exit`, `logout` or CTRL+D (but stay on Ela) and use the command `passwd` to change your password. Log back to Eiger with the new password.

Note (more understandable after visiting first Exercises 5 and 6): on Mac, Linux and Windows10 systems, one can also modify `~/.ssh/config` file by adding (if file `config` does not exists within `.ssh` directory, create it using `touch config` inside `.ssh` directory)

```
# replace cscsusername with your CSCS username
Host ela
  Hostname ela.cscs.ch
  User cscsusername
  ForwardX11 yes
  ForwardAgent yes
```

Then, one can type only `ssh ela` instead of `ssh -X cscsusername@ela.cscs.ch`.

Finally, one can even configure the `ssh` to log in directly from your computer to Eiger without logging first to Ela. To do this, you should add the following lines to your `config` file:

```
# replace cscsusername with your CSCS username
Host eiger
  Hostname eiger.cscs.ch
  User cscsusername
  ForwardX11 yes
  ForwardAgent yes
  ProxyJump ela
```

With this, you can type `ssh eiger` directly from your terminal and ssh will perform the jump via Ela front end for you.

Exercise 3 [Graphical interface]

In this exercise, we construct a simple figure using `Python`, save the output and view it in the graphical session. Log in to Eiger and load Python module. In addition, install

required Python packages (for this, only `matplotlib` should suffice) using the following two commands:

```
module load cray
module load cray-python
pip install matplotlib
```

To produce plot, run python script `plot_functions.py` located in `/capstor/store/cscs/uzh/uzh8/ESC401/exercise1/` directory as follows:

```
python /capstor/store/cscs/uzh/uzh8/ESC401/exercise1/plot_functions.py
```

Try to view the file you produced (e.g. command `display` might be useful, do not forget that you need to have X11 forwarding configured).

Exercise 4 [Automatic login]

As introduced during the lecture, we are going to setup passwordless ssh-ing. For this you need to generate a personal authentication key on your local system, place the public part of the key on the server and keep the private part on your system. This can be done with `ssh-keygen`. Here is what you have to do on your local computer (laptop):

(i) Check in `~/.ssh` if the keys already exist on your computer. If the files `id_rsa` and `id_rsa.pub` are present, you can skip the following step.

(ii) Create public and private keys using ssh-key-gen:
`ssh-keygen -t rsa`
Don't set a passphrase as this is the most straightforward way to access remote client without typing any password.

(iii) Create a directory `~/.ssh` on the remote system. You can do this from your local computer using ssh:
`ssh user@remote-host mkdir -m 700 .ssh`
The command `mkdir` will executed on the remote host. The option `-m 700` sets the permissions so that only the user can access the directory.

(iv) Copy the public key to remote-host:

```
ssh-copy-id -i .ssh/id_rsa.pub user@remote-host
```

which appends `.ssh/authorized_keys` file with the public key (or creates this file if not existing). Alternatively, one can use command:

```
cat .ssh/id_rsa.pub | ssh user@remote-host "cat >> .ssh/authorized_keys"
```

This command will append the key if `.ssh/authorized_keys` already existed.

(v) Try your new settings!

Repeat the procedure between Ela and Eiger (if necessary).

Exercise 5 [Getting accustomed to the UNIX shell]

Once you set-up the connection to Eiger, you can start exploring the environment. All interaction with the system happens via the shell, which will interprete your commands and call the appropriate programs. One very useful programs is `man <command>`, which is short for manual. With this program one can find informations about any program installed (which has `man` pages). You can use `apropos` to research for a keyword in the short description of all man pages. This is handy if you don't know the name of the command you need.

- Using `man` learn about `top` and `ps` commands. Can you describe what they do? What does the flag (modifier) `-u` do? What is the result of executing `top -u <your login>` and `ps -u <your login>`? How can you sort the result of `top`?
- Execute `sleep 10`. What is the effect of your command?
- What do the following commands do and how can you use them to navigate around? Commands: `pwd, ls, cd, rm, cp, ln`
- ***Hand-in:*** In `/capstor/store/cscs/uzh/uzh8/ESC401/exercise1/` directory, you can find a file named `data.txt`. What does it contain? (Hint: try to use the Python file `plot_data.py` located in the same directory and revisit Exercise 3). Try to copy the plot produced by a python script to your computer (commands `scp` or `rsync` might be useful).
- Execute `yes`. What do you see? What does this command do? You can break its running by pressing CTRL+C or suspend it with CTRL+Z. Do the latter and find this job's ID with `ps`. Unresponsive (and other) jobs can be 'killed'. Learn what is the effect of running `kill -9 <job id>` and kill the suspended `yes` job. What other modifier could you use?
- Jobs which were executed in terminal and require user's intervention will block the terminal (no other command can be executed). Run `man cp &` and `bc &`. What did you achieve? Is `man` running or not? What is the ID of the job? What about `bc`? What output do you get when you execute `jobs`? Jobs can be brought to foreground or send to background with `fg` and `bg`, respectively. Kill `man` WITHOUT using its ID number and bring `bc` to foreground. Which commands did you execute? (Hint: use `%<job's number>`)

- What does the `echo` function do? What do you get when calling `echo Hello world!`, `echo $USER`, `echo $SHELL`, `echo $HOST`?
- Use command `df` to find out how much space you have in your home directory.
- ***Hand-in:*** Log out of Eiger and use `scp` to copy the image from exercise 3 to your local system.

Exercise 6 [Text editors]

The `vim` editor, standing for vi improved, is included in most UNIX systems and can be used with a command line interface. You can find an interactive tutoral at http://openvim.com if you want to learn the basics.
Create a file with `touch` and edit it with `vim` or any other terminal-based text editor of your choice (`emacs`, `nano`). Write at least 5 lines of text and save it. Copy the file (`cp`) and count number of words and lines with `wc`

- ***Hand-in:*** What did you get? Append the copied file to the `data.txt` file from Exercise 5 (multiple ways possible - suggested: `cat` and redirecting)