Introduction & Guidelines exercise sessions Technisch Wetenschappelijke Software/ Scientific Software

Emil Løvbak & Pieter Appeltans *October 2, 2020*

1 Introduction & Evaluation

During the guided exercise sessions, we will study different aspects of scientific software, such as finite-precision arithmetic, debugging, and code performance. Eight guided exercise sessions make use of Fortran and C++ to apply the concepts of scientific computing taught in the lectures. These guided exercise sessions serve as a springboard for the five homework assignments and attendance is therefore **highly recommended**. Two graded exercise sessions are devoted to Matlab and combining programming languages. As these exercise sessions are graded, attendance is **mandatory**.

The evaluation of this course is based on the homework assignments and graded exercise sessions. These serve as a basis for the oral exam in January, where you will be asked to explain your code and answer general questions about the course material. During this exam, you must show what you learned from the lectures, the exercise sessions, the assignments and the provided feedback.

As the first assignment starts soon, it is important that you start working on this course from **week one!**

The assignments should be made **individually** (for more information on what is allowed see Section 2.5).

Remark 1: A student who obtains 6/20 or less for one third or more of the assignments cannot pass the course.

Remark 2: There is no possibility for a reexamination in the August/September examination period. Spreading the work over the academic year and the September examination period is also not possible. Exceptions to this rule can however be allowed, if sufficiently motivated.

2 Guidelines

2.1 Corona

Regularly check the universities guideline regarding education during the Corona pandemic (*Dutch:* `https://www.kuleuven.be/onderwijs/student/onderwijs-in-coronatijd` - *English:* `https://www.kuleuven.be/english/education/student/2020-2021_education`) and the frequently asked questions (*D:* `https://www.kuleuven.be/coronavirus/veelgestelde-vragen#ONDERWIJS` - *E:* `https://www.kuleuven.be/coronavirus/english/FAQ#EDUCATION`). Please adhere at all times to the code of conduct of the university (*D:* `https://www.kuleuven.be/studentenvoorzieningen/thuis-aan-kuleuven/afsprakenkader` - *E:* `https://www.kuleuven.be/english/studentservices/at-home-at-kuleuven/code-of-conduct`).

Important: If you are unable to attend one of the exercise sessions (guided or quoted), do not hesitate to contact the teaching assistants such that an (online) alternative can be worked out.

2.2 Discussion forum

If you still have questions after an exercise session or if you get stuck on a homework assignment, do not hesitate to post a question on the discussion forum on Toledo. This forum is regularly

monitored by the teaching assistants.

2.3 Guided exercise sessions

You should complete the preparation posted on Toledo **before** the start of each exercise session. From past experience, we have noticed that unprepared students struggle during the exercise session and are often unable to complete all exercises, missing important information as a result.

2.4 Handing in tasks

For the assignments and the quoted exercise sessions, you should upload the necessary source file(s) and a report (if specified) on Toledo. The deadline for submitting will be communicated well in advance. **These deadlines are strict,** late submissions will be sanctioned.

Before submitting your work, check that

- your name is in the report and in comments at the top of each file;
- you answered all questions in a clear and understandable way;
- you mentioned in comments at the top of the main file which compiler(s) you used to test your program and the compiler command(s) that compile(s) your program (an additional `Makefile` is also an option);
- you correctly labelled the axes of your figures and added (if necessary) a legend;
- your figures are readable;
- your assignment satisfies the (style) requirements specified by the assignment (eg. reports should be PDF files and not Word, etc.);
- you did not include unnecessary files such as *.o files, *.mod files, *.out files, old versions of the code or executables;
- you included how much time you spent on the task in the report or in comments at the top of the main file (this allows us to evaluate the workload of the assignments).

After each home work you will receive individual feedback: take it into account when making the next tasks and when preparing your exam.

Remark 3: For the students of the Dutch course (Technisch Wetenschappelijke Software) it is allowed to write your report in Dutch.

2.5 Fraud & Plagiarism

All tasks and quoted exercise sessions are graded, and the exam rules therefore apply. In the following we describe what is and is not allowed in terms of cooperation.

The solution and/or report and/or program code that are handed in, have to fully be the result of work you have performed **yourself**. You can of course discuss with other students, in the sense that you may talk about general solution methods or algorithms, but the discussion cannot be about specific code or report text that you are writing, nor about specific results that you wish to hand in. If you talk with others about your tasks, this can **NEVER** lead to you being in possession of a whole or partial copy of the code or report of others, regardless of the code or the report being on paper or available in electronic form, and independent of who wrote the code or report (fellow students, possibly from other study years, complete outsiders, internet sources, etc.). This also encompasses that there is no valid reason at all to pass your code or report to fellow students, nor to make this available via publicly accessible directories or websites.

Every student is responsible for the code and the work he or she hands in. If there are any

doubts during the assessment of a task about the fact whether this is self made (e.g. similar code, graphs, text or results), the student will be asked to give an explanation for this. If this does not eliminate the doubts, this will be reported as an irregularity, as foreseen in the education and exam regulations (see `http://www.kuleuven.be/education/regulations/index.php`).

Give references for everything you submitted that is not in the course material or that is not 'generally known'. Certainly refer to original code if you adapted/used it (for a non-essential part of the task !).

P.S. Plagiarism detection software may be used to detect problems.

3 Preliminary assignment

Complete the following exercises before the first exercise session. It is advised that you work in the departmental PC rooms, either physically or remotely (see Exercise 5). If you have any problem or question during this assignment, do not hesitate to post them on the discussion forum on Toledo.

Exercise 1: Useful terminal commands

On the departmental computers, you will work with Ubuntu. To log in use your KU Leuven student account (r-number and password you use for Toledo). In the exercise sessions we will often use the Ubuntu command line interface terminal (you can open the terminal by pressing Ctr-Alt-T). If you have not worked with the terminal before, familiarise yourself with the following commands (for more information see `https://help.ubuntu.com/community/UsingTheTerminal`). If you are already accustomed with the terminal you can skip this exercise.

- `$ cd path/to/directory`: navigates to the specified path.
- `$ cd ..` : navigates up one directory level
- `$ cd` $\sim$: navigates to the home directory.
- `$ ls` : lists the files in the current directory.
- `$ pwd` : prints the path of the directory you are currently working in.
- `$ mv path/to/existing_file path/to/new_location` : moves file to new location.
- `$ cp path/to/existing_file path/to/new_location` : copies file to new location.
- `$ mkdir dir` : makes a new directory with the name `dir`.
- `$ cat path/to/file` : outputs the content of file to the standard output.
- `$ top` : shows a real-time view of the running processes.
- `$ clear` : clears the standard output.
- `$ exit` : closes the terminal.
- $\uparrow$ (up arrow on key board) : scrolls through the commands you've entered previously.
- Tab : autocompletes command or filename.
- Ctrl-c : interrupts the execution of a program.

Exercise 2: Editors

To write code, you will need to use a code editor. For this course, we will use Visual Studio Code; for more information on how to set-up this editor on your personal machine see the document "Home set-up for Technisch Wetenschappelijke Software/Scientific Software".

Exercise 3: Setup and activating Fortran compilers

To be able to use the Fortran compilers on the departmental computers, you need to activate them first for your account. To do so, follow the following steps.

1. Log in on a machine in the departmental PC rooms.
2. Open the terminal (Ctr-Alt-t).
3. Open the file ∼/.options.sh (for example via `$ nano ∼/.options.sh`).
4. Add `CS_FORTRAN=yes` to the end of this file.
5. Save and close the file (Ctr-x in `nano`).
6. Restart the terminal.

Exercise 4: Write, compile and run your first program

Now we are ready to write our first Fortran program.

1. Create a new file and save it as `hello_world.f95`.
2. Add the following lines to this file and save it again.

```
program hello_world
print *, "Hello, world!  This is my first Fortran program."
end program
```

3. Compile your program (for more information see Toledo-Exercise sessions-Documentation-Compilation.pdf):
   (a) Open a new terminal window (Ctr-Alt-T).
   (b) Navigate to the directory where you stored your file (use the commands from Exercise 1).
   (c) Compile your program to an object file: `$ gfortran -c hello_world.f95`.
   (d) Link your program: `$ gfortran -o hello_world hello_world.o`.
4. Now you are ready to run your program: `$ ./hello_world`. The text "Hello, world! This is my first Fortran program." should appear on the standard output.

Exercise 5: SSH

If you want to work on the departmental machines from outside the department, you can use SSH. The following tutorial will guide you through the steps to establish a SSH connection to the departmental computers [1].

Part 1 (only first time) The following steps will be slightly different depending on the operating system on your personal machine. Windows 10 has introduced support for OpenSSH, which should work out of the box. Older versions of Windows will require extra tools, as you can see below.

1. Generate a public-private ssh key pair on `https://www.cs.kuleuven.be/restricted/computerklas/keys.php`. (**You might have to wait one day until your key is activated on the departmental servers.**).

[1] This tutorial is based on the information found on `http://dietercastel.com/2013/03/02/ssh-to-computer-science/`, `https://system.cs.kuleuven.be/cs/system/wegwijs/computerklas/internet/index-E.shtml` and `https://system.cs.kuleuven.be/cs/system/wegwijs/computerklas/machines/index-E.shtml`

Linux (personal machine)

2. Download the files id_rsa, id_rsa.pub, id_dsa and id_dsa.pub and save them in the folder ∼/.ssh.
3. Use `chmod 700 id_rsa` and `chmod 700 id_dsa` to change the access rights of these files.

Mac (personal machine)

2. Install xquartz from `https://www.xquartz.org/`
3. Download the files id_rsa, id_rsa.pub, id_dsa and id_dsa.pub and save them in the folder ∼/.ssh. **Make sure that the files do not have `.txt` as extension!**
4. Use `chmod 700 id_rsa` and `chmod 700 id_dsa` to change the access rights of these files.

Windows (newer personal machine)

2. Download the files id_rsa, id_rsa.pub, id_dsa and id_dsa.pub and save them in the folder `.ssh` in your home folder (most likely `C:\Users\<username>`). If the folder `.ssh` does not yet exist then create it. **Make sure that the files do not have `.txt` as extension!**

Windows (older personal machine)

2. Download the file id_rsa and save it somewhere on your machine.
3. Download MobaXterm (`https://mobaxterm.mobatek.net/`) and install it.
4. Open MobaXterm.
5. Click on 'Session' (top left).
6. Click on 'SSH'.
7. Remote host: machine.cs.kotnet.kuleuven.be with machine the name of a PC in the departemental PC rooms.
8. Username: your student number.
9. In network settings, check "Connect through SSH gateway". Servername: `st.cs.kuleuven.be`, Username: your r-number and key file the id_rsa file you downloaded before.

Part 2 (every time you log in)

Note: You might have to wait one day until your key is activated on the departmental servers.

1. Find an available PC. For Linux, Mac and newer Windows computers execute the following command in the terminal/ Windows Powershell:
`ssh -N -L 10080:mysql.cs.kotnet.kuleuven.be:80 r0000000@st.cs.kuleuven.be`
with r0000000 your student number, enter your keyphrase and open `http://localhost:10080/` in your browser.

Linux

2. Open the terminal on your own machine.
3. Connect to your PC of choice:
`ssh -X -J r0000000@st.cs.kuleuven.be r0000000@machine.cs.kotnet.kuleuven.be`
with r0000000 your student number and machine the name of a machine in the departmental PC rooms.
4. Enter your ssh key phrase (twice).
5. Now you are connected to a machine in the departmental PC rooms.

Mac

2. Start xquartz.
3. Connect to your PC of choice:
   `ssh -X -J r0000000@st.cs.kuleuven.be r0000000@machine.cs.kotnet.kuleuven.be`
   with r0000000 your student number and machine the name of a machine in the departmental PC rooms.
4. Enter your ssh key phrase (twice).
5. Now you are connected to a machine in the departmental PC rooms.

Windows (newer machines)

2. Open a Windows Powershell window.
3. Connect to your PC of choice:
   `ssh -X -J r0000000@st.cs.kuleuven.be r0000000@machine.cs.kotnet.kuleuven.be`
   with r0000000 your student number and machine the name of a machine in the departmental PC rooms. If this does not work, use instead
   `ssh -o "ProxyCommand ssh.exe -W %h:%p r0000000@st.cs.kuleuven.be" r0000000@machine.cs.kotnet.kuleuven.be`
4. Enter your ssh key phrase (twice).
5. Now you are connected to a machine in the departmental PC rooms.

Windows (older machines)

1. Open MobaXterm.
2. Click on the connection you created before.
3. Enter your keyphrase (twice).
4. You should now be connected to a departemental PC.

Part 3 using VS Code

To connect to a departemental PC using VS Code on your personal machine, see the document "Home set-up for Technisch Wetenschappelijke Software/Scientific Software".

Exercise 6: GIT (optional)

It could be useful to use a version-control system, such as GIT, which allows you to back up your code and restore old versions. You can also use it to compare different versions of your code. For more information see:

- `https://education.github.com/pack`
- `https://bitbucket.org/product/`
- `https://git-scm.com/`
- `https://www.atlassian.com/git`

If you want to use GIT or another version control system, familiarize yourself with it before the first exercise session. **Make sure that the repository with your code is private to be inline with the guidelines in Section 2.5.**