

**Computer lab**  
**Numerical Methods for Thin Elastic Sheets**  
Summer term 2013  
Prof. Dr. M. Rumpf – B. Heeren, R. Perl

## **Technical introduction**

### **Log in**

Log in on the terminal computer using the username `prakt0i` (with  $i = 1, \dots, 6$ ) and the well-known password or the username/password provided on the screen. It might be reasonable to create a personal folder name in the home directory where you can store everything mentioned subsequently. Furthermore, you will get your personal account details for accessing the repository from the tutor.

### **Download the source code**

The source code we are going to work with is stored in a remote repository. Any interaction is realized by means of a revision control tool named *Mercurial*. First, you have to download the existing code files from this repository, i.e. create a local *clone* of it. Therefore:

- copy `hgrc.txt` (provided on the website) to your home directory, rename to `.hgrc` and edit this file by replacing `[testperson]` by your name (this will be the name that will appear as author for any code changes committed by you)
- log in on `https://source-numod.ins.uni-bonn.de/hg/` using your personal account details, go to `praktSS13` and copy the "clone url"
- open the graphical user interface *TortoiseHG*, e.g. by entering "thg" in your terminal
- go to `File->Clone Repository` and enter the url for "Source"
- choose a name for the local *source directory* "Destination" (e.g. `home/name/praktSS13/`) and clone the repository entering your personal password again (the source code is now in your source directory)

## Compiling with CMake

To compile the source code proceed as follows:

- create a *build directory* where the binaries (i.e. the executable files) are to be stored (recommendation: on the same level as your source directory, e.g. `home/name/build/`)
- copy the file `cmake.selection.default` from the destination to the build directory and rename to `cmake.selection`
- open the graphical user interface for *cmake*, e.g. by entering "cmake-gui" in your terminal
- enter your source directory (first line) and the build directory (second line)
- push configure (and use default settings)
- to get rid of errors, untick `USE_CUDA` (and maybe untick `USE_GRAPE`, `USE_OPENMESH`, `USE_IPOPT`, `USE_AHMED` and `USE_CIMG` to get rid of warnings)
- push configure again and push generate
- open a terminal, browse to your build folder and type `make`

First test example:

- create a directory "results" (on the same level as your build directory) and go to `/examples/linearTriangleFE` in your *build* directory
- try to execute `./linearTriangleFE` with a suitable parameterfile (e.g. use `/examples/linearTriangleFE/linearTriangleFE.par` in your *source* directory)
- have a look at the results in your results directory

## Create your own project

To avoid global conflicts at the beginning we suggest that everyone creates his own project folder:

- make a new directory with your name, i.e. `/projects/name`, in your *source* directory
- as a first example, copy `../examples/linearTriangleFE/linearTriangleFE.cpp` to that directory and rename it (e.g. to `test.cpp`)
- add `/projects/name` in your `selection.default` (in your *build* directory)
- call `make` and try to execute your personal main program (remove compile errors)
- open *TortoiseHG*, choose "praktSS13" on the left hand side and click on "Working directory" in the top line on the right hand side
- add your personal `cpp`-file (should appear somewhere in pink color at the bottom) by right-click and "add"
- commit (with some commit message) your changes (**Caution:** everything ticked will be committed!! Hence untick everything that is not in your personal project!!) and push to repository

## Documentation

You can create a doxygen documentation as follows:

- type `make docall` in your *build* directory
- open `/doc/all/index.html` (in your *build* directory!) in a browser