Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

universitätbonn

Practical Lab
Variational Methods and Inverse Problems in Imaging

Summer term 2014

Prof. Dr. M. Rumpf – A. Effland, B. Geihe, S. Simon, S. Tölkes

Technical issues

Time and place

The weekly meeting will take place on Fridays at 12 o'clock in our terminal room Z2.067. If need be we may change to Tuesdays at 16 o'clock.

Office hours will be from Monday to Thursday from 13:30 to 15:00; on Mondays with Stefan Simon in room Z2.065, on Tuesdays with Benedict Geihe in room 2.047, on Wednesdays with Sascha Tölkes in room 2.047 and on Thursdays with Alexander Effland in room Z2.064.

Web page

The web page to this practical lab is located at:

`http://numod.ins.uni-bonn.de/teaching/ss14/practicalLab/`

Here you can find problem sheets and an up to date documentation.

Log in

Log you can log into one of the computers in the terminal room using the user name `prakt0i` (with $i = 1, \dots, 6$) and the well-known password. 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*. It facilitates colloborating with other group members and allows for easy roll back in case things break. First, you have to download the existing code files from this repository, i.e. create a local *clone* of it. Therefore:

- copy `.hgrc` (provided on the website) to your home directory
- edit this file to replace <testperson> by your own name (this name will appear as the author of any code changes committed by you)

- go to `https://source-numod.ins.uni-bonn.de/hg/` using a web browser and supply your personal account details. Click on `praktSS14` and copy the "clone URL", it should be `https://<username>@source-numod.ins.uni-bonn.de/hg/praktSS14`
- you have two options now: either run *Mercurial* via the command line or by using the graphical user interface *TortoiseHG*.

  For the command line:

  - open a console
  - navigate to the folder where you want the code to be downloaded
  - type `hg clone <URL>` where `<URL>` is the "clone URL" you obtained earlier
  - enter your password

  For *TortoiseHG*:

  - open *TortoiseHG*, e.g. by entering `thg` in your terminal
  - go to `File->Clone Repository` and enter the "clone URL" you obtained earlier as "Source"
  - choose a name for the local *source directory* "Destination" (e.g. `<name>/praktSS14/`) and clone the repository entering your personal password again. The source code should now be 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. `<name>/build/`)
- copy the file `cmake.selection.default` from the source directory to the build directory and rename it 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)
- click `configure` (to use default settings)
- click `generate`
- open a terminal, navigate to your build folder and type `make`
- if everything went smoothly, try to execute the first example `build/projects/introduction/exercise01`

Create your own project

To avoid global conflicts we suggest that every project creates its own project folder:

- create a new directory with the project name, i.e. `source/projects/<name>` in your *source* directory

- as a first example, copy `../introduction/exercise01.cpp` to the new directory and rename it if you like
- add `projects/name` in your `selection.default` (in your *build* directory)
- call `make` and try to execute the new program To upload the new source file using the command line:
  - go to your *source* directory
  - type `hg add projects/<name>/<newfile.cpp>`
  - type `hg commit projects/<name>/<newfile.cpp> -m "first example file"` (where `-m` indicates a comment)
  - type `hg push` to do the upload

  To use *TortoiseHG*:
  - open *TortoiseHG*, choose "praktSS14" on the left hand side and click on "Working directory" in the top line on the right hand side
  - add the new file (should appear somewhere in pink color at the bottom) by right-click and clicking "add"
  - commit (with some commit message) your changes.
  - click on `Repository->Synchronize->Outgoing`, verify that the right changes are going to be uploaded, and click "Accept".

Documentation

You can create your own local `doxygen` documentation as follows:

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