

# SOFTWARE ENVIRONMENT

JOHAN JANSSON

## 1. COMPUTER ENVIRONMENT

The course provides computer resources for working with the software content of the course modules.

1.1. **Computational server.** Available is the computer `u-shell.csc.kth.se` which has the necessary tools (Python and FEniCS) installed and supported. You can log into this computer with an SSH client and run Python and FEniCS programs.

1.2. **Lab computers.** The lab computers are available as terminals from where you can SSH into `u-shell.csc.kth.se`. You can also run ParaView on the lab terminals and visualize your functions.

1.3. **Python.** Python is a powerful and understandable programming language available as free software (full source code available and re-distributable) from [python.org](http://python.org).

We assume a basic computer science and programming course as background. Available is a [tutorial](#) which gives an introduction to the language. Also available is a book which in addition gives an introduction to some of the computer science concepts available in Python.

Python will be used in the course to write programs using the FEniCS system, which is available as a Python module.

The course will only require a very basic understanding of the Python language.

You need to know:

- how to start Python
- how to execute a Python program
- basic debugging (what basic error messages mean)

Syntax for:

- declaring variables/objects
- calling functions
- declaring functions
- declaring classes and inheritance (limited use)
- basic input/output

1.4. **FEniCS.** FEniCS provides free software for automation of computation. The full source code is available and re-distributable under the GNU General Lesser Public License from [fenics-project.org](http://fenics-project.org).

The course will give you an understanding of the algorithms implemented in FEniCS, but you will not need to re-implement FEniCS. You will instead be able to use the algorithms as building blocks when solving problems given in the course modules. Since FEniCS is free software you will be able to inspect the implementation if you don't understand how an algorithm works and use or extend FEniCS for future projects.

1.4.1. *Overview.* FEniCS provides an abstraction to concepts needed for FEM discretization such as linear algebra and mesh representations, and also provides representations of FEM data structures and algorithms such as weak forms and assembly.

In the course we will use the following abstractions that FEniCS provides:

**Mesh:**

- Iterating over mesh entities (vertex, facet, cell)
- Mesh operations (refinement)

**Linear algebra:**

- Tensor representation and operations (vector, sparse matrix)
- Linear system solvers (Krylov, LU)

**Function:**

- Solution and coefficient representation (discrete, user-defined)

**Form:**

- Bilinear and linear weak forms (for representing a PDE)

**Assembly:**

- For assembling a matrix or vector from a bilinear or linear form

1.5. **ParaView.** ParaView is a mature, understandable visualization application available as free software (full source code available and re-distributable) from [paraview.org](http://paraview.org).

When developing a numerical algorithm or a mathematical model, it can be useful to look at a graphical representation of solution/coefficient functions to be able to directly see incorrect results or be able to estimate quantities using just your eyes. Together with computed values such as norms of quantities this can lead to a rapid development.

ParaView is part of the computer environment of the course (installed on the lab computers). It can be started by typing:

```
paraview
```

1.6. **Getting started (Mac and Linux).** This is a list of steps for logging into the computer environment and getting started with FEniCS:

- (on lab/home computer) login to computational server u-shell:  
`ssh u-shell.csc.kth.se`
- (on server) download a sample program:  
`wget http://www.csc.kth.se/utbildning/kth/kurser/DN2260/fem12/demo.py`
- (on server) run the program using Python  
`python demo.py`
- (on lab/home computer) start ParaView to look at solution  
`paraview`
- (on lab/home computer) edit source file `demo.py` and make changes/run again

1.7. **Getting started (Windows).** Download the [Putty](#) SSH program. Connect to `u-shell.csc.kth.se` and follow the instructions above from the second point. Transfer the output files using for example [WinSCP](#) to your laptop and visualize them using [ParaView](#).