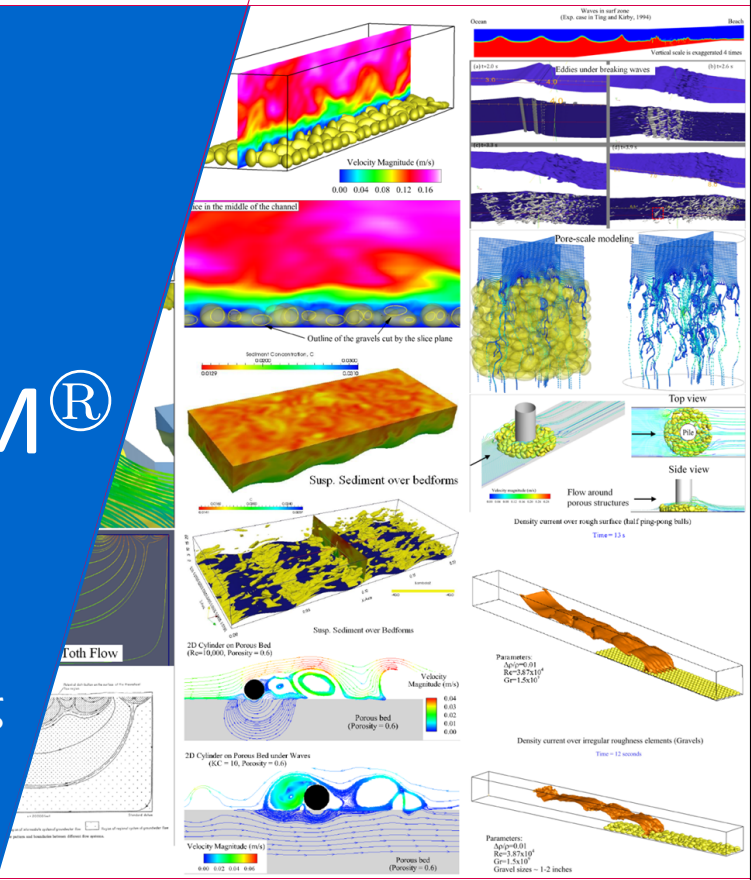
PENNSTATE

1855

CE597D Special Topics, Fall, 2014

Chapter 2, Part 2: Installation and Initial Browsing of OpenFOAM®



Xiaofeng Liu, Ph.D., P.E.
Assistant Professor
Department of Civil and Environmental Engineering
Pennsylvania State University
xliu@engr.psu.edu

Outline

Level of users

- Beginner level:
  - A brief introduction of OpenFOAM®
  - Demonstration of OpenFOAM® usages
  - Installation of OpenFOAM®
  - Hands-on exercises for the basics
  - Detailed walk-through of the code
  - ...
- Intermediate and advanced levels (later):
  - Demonstration of OpenFOAM® development
    - Solver
    - Boundary conditions
    - Tools
  - Hands-on exercise
  - Discussion of specific applications

What we have done so far:

- *A brief introduction of OpenFOAM®*
- *Sample applications of OpenFOAM®*
- Installation of OpenFOAM®
  - Brief introduction of Linux system
  - Installation of OpenFOAM®
  - Test the installation
  - Initial browsing of the OpenFOAM® code
- Hands-on exercises for the basics
- Detailed walk-through of the code

Brief introduction of Linux system

Starting off with Linux system

- Has always been a command-line based system
- Currently very good GUI support
- Many distributions of Linux: Ubuntu, Centos, SuSe, Gentoo, Fedora, Debian, etc.
- Distributions differ in the number of customizations, customer support, package management, release cycles etc.
- Common Linux directory structure

Brief introduction of Linux system

Linux directory (file) system structure

- /home - user home directories
- /usr - sources, include files, shared files, libraries, local installations ...
- /lib - libraries
- /bin - executables
- /opt - optional software
- /etc - configuration files, init scripts ...
- /boot - kernel image
- /proc - kernel data structures
- ...

```
[xliu@xliuws0514 /]$ cd /
[xliu@xliuws0514 /]$ tree -C -L 1
.
├── bin
├── boot
├── dev
├── etc
├── home
├── lib
├── lib64
├── lost+found
├── media
├── misc
├── mnt
├── net
├── opt
├── proc
├── root
├── sbin
├── selinux
├── srv
├── sys
├── tmp
├── usr
└── var

22 directories, 0 files
[xliu@xliuws0514 /]$
```

Figure: Directory structure of a typical Linux system

Brief introduction of Linux system

Interaction with the Linux system:

- Graphical User Interface (GUI)
  - Easy and straightforward to use
  - Improved over the years
  - However, sometime slow and not efficient.
    - Example: find the size of each sub-directory in your home directory and save them to a file. It is very easy to do using command:

      ```
      [xxx@xxxx ~] du -sh * > output.txt
      ```
- Command Line Interface (CLI), i.e., shell
  - Was the only way to interact with a computer in the old days
  - CLI or shell: a program that reads your commands from the keyboard, process and interpret them, and then pass them along to the operating system to execute.
  - Popular shells: bash (Bourne Again SHell), tchsh, ksh, zsh, etc.
  - How to find out what shell you are using:

    ```
    [xxx@xxxx ~] echo $SHELL
    ```

    or

    ```
    [xxx@xxxx ~] ps -p $$
    ```
  - We will use bash for this course.

Brief introduction of Linux system

bash shell:

- Most popular
  - history
  - command line editing
  - rich customization
- Run '`bash --version`' to find out the version of your bash
- Basic commands
  - cd ls pwd
  - echo export
  - alias
  - cat head tail touch
  - top ps kill
  - find locate which
  - grep more less
- Commands can be stored in a bash script file to be executed all at once. bash script language:
  - define variables
  - loops, switch, for
  - define functions
  - ...

Brief introduction of Linux system

bash shell:

- Pipes, Redirection, Indirection
  - String together a bunch of simple commands to achieve a complicated task
  - `cmd1 | cmd2 | ....`
  - Redirect the output: `cmd1 > some_file`
  - Specify the input file and output file: `cmd1 < input_file > output_file`
  - Append the output to an existing file: `command >> output_file`

Brief introduction of Linux system

bash shell:

- Process management
  - ‘top‘ command: An interactive tool to observe load, mem usage, cpu usage, user, command executed, send signals, process state
  - ‘ps‘ command: show processes on a machine
  - ‘kill‘ command: kill certain process
  - foreground, background processes: `cmd &`

Brief introduction of Linux system

bash shell:

- Process management
  - ‘top‘ command: An interactive tool to observe load, mem usage, cpu usage, user, command executed, send signals, process state
  - ‘ps‘ command: show processes on a machine
  - ‘kill‘ command: kill certain process
  - foreground, background processes: `cmd &`
- Environmental variables:
  - They provide a way to change the behavior of the software and the Linux system itself.
  - A few well-known environmental variables: `$PATH`, `$SHELL`, `$HOME`
  - You can define your own: For example, `WM_PROJECT` is defined as `OpenFOAM` by the following line

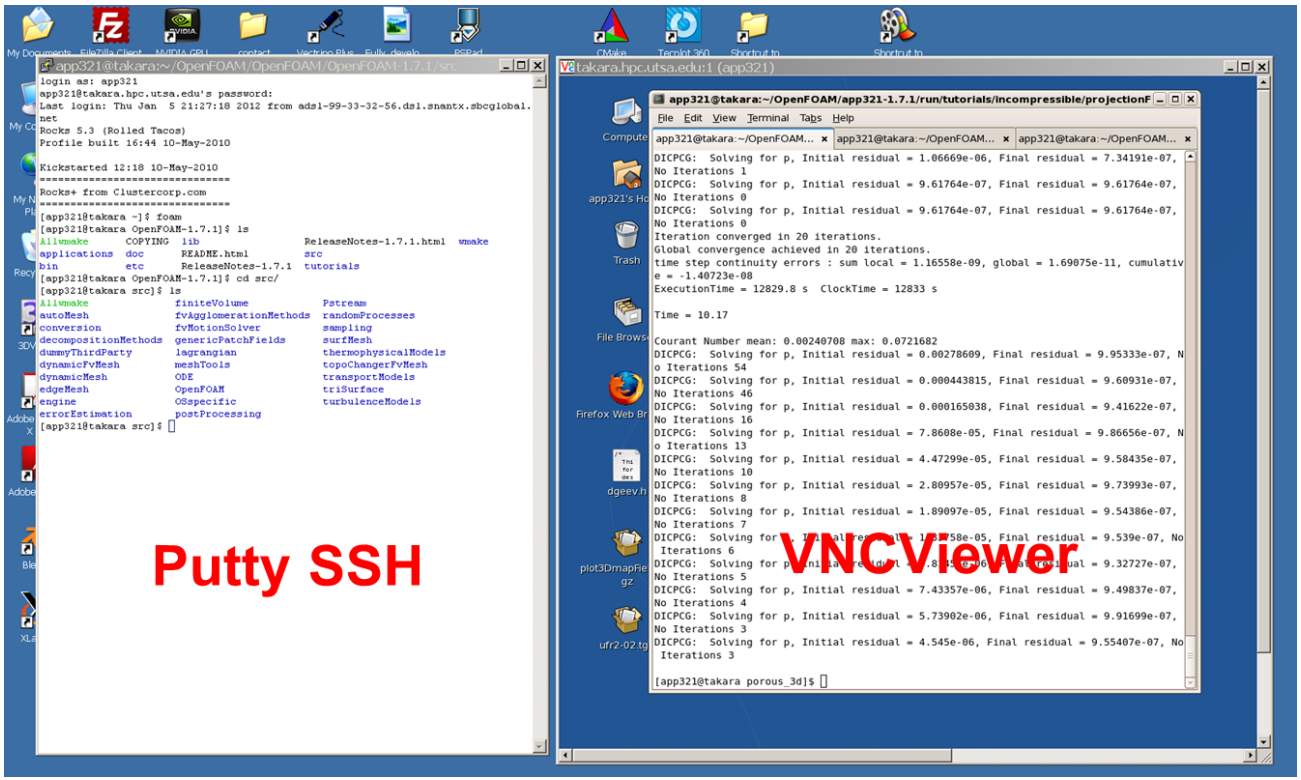
    ```
    export WM_PROJECT=OpenFOAM
    ```

  - Type ‘`export`‘ or ‘`env`‘ to see all the currently defined environmental varialbes
- You can configure your bash shell with the ‘`.bash_profile`‘ and ‘`.bashrc`‘ script files in your home directory:
  - ‘`.bash_profile`‘: read once when you login
  - ‘`.bashrc`‘: read every time a shell is started

Brief introduction of Linux system

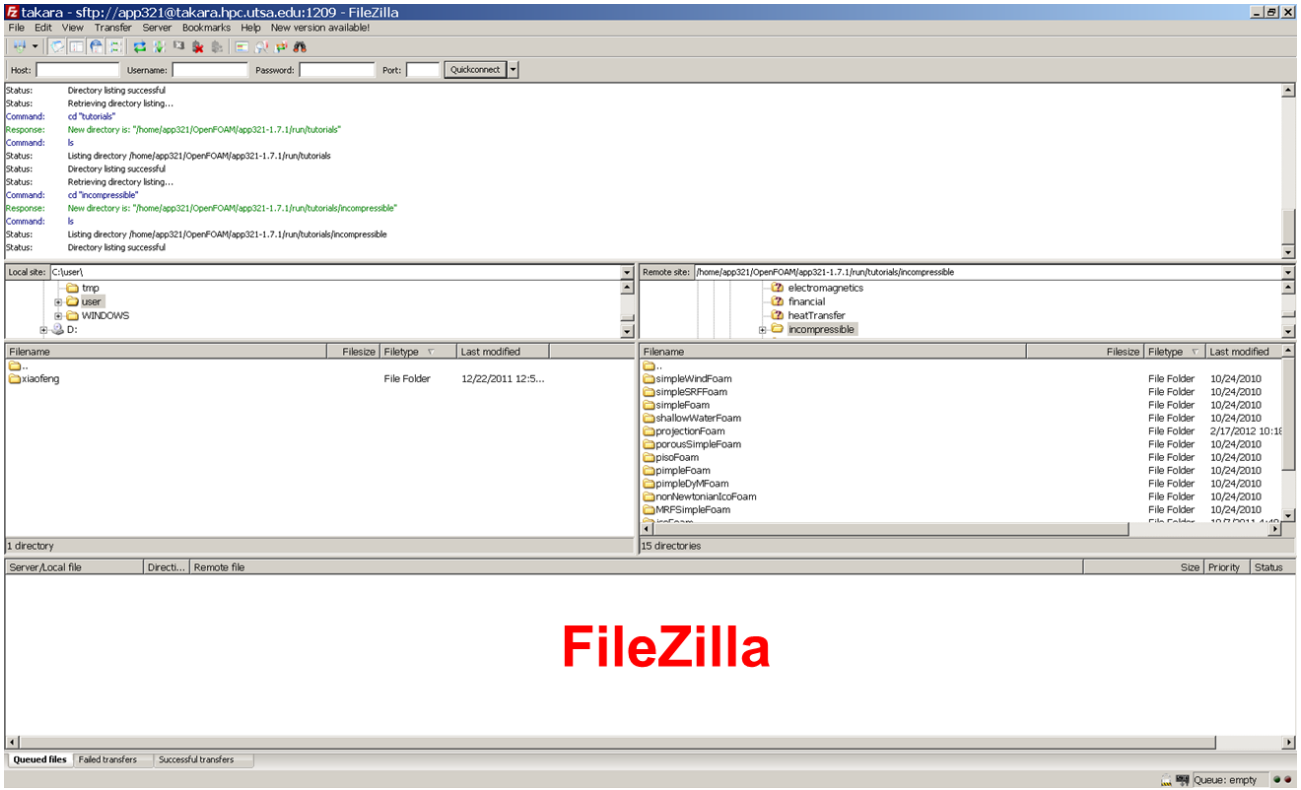
Remote login to a Linux computer:

- SSH: I use putty on my Windows machine
- SCP: securely copy files from machine to machine; I use the freeware FileZilla (SCP with a GUI)
- VNCServer/Viewer: I configure a VNC service on my Linux machine and run RealVNC client on my desktop
- On PSU hammer machine, you need to use *Exceed OnDemand*



Brief introduction of Linux system

Remote login to a Linux computer:



Brief introduction of Linux system

- Linux programming environment
  - Languages supported: C, C++, Fortran etc.
  - Debugger: gdb
    - compile with -g option
    - Breakpoints, watch, conditional breakpoints
- Text editors:
  - `gedit`: GUI, very similar to Notepad in Windows, but not efficient
  - `vi`: One of the oldest editors, simple and fast
  - `emacs`: Extensible editor, very powerful but complicated
- Compilation using `make`
  - The idea is to build and manage your (large) computer code automatically.
  - GNU make, or simply gmake: read `Makefile`
  - OpenFOAM® uses `wmake`: similar to gmake. More on this later.
    http://www.openwatcom.org/index.php/Using_wmake

Brief introduction of Linux system

- Good reference books for Linux systems
  - *Beginning the Linux Command Line*
  - *A Practical Guide to Linux Commands, Editors, and Shell Programming*
- Good books for C++ programming
  - *Thinking in C++: Introduction to Standard C++*
  - *Effective C++: 55 Specific Ways to Improve Your Programs and Designs*

Installation of OpenFOAM®

- Get OpenFOAM®
  - Download from the "official" website
    http://www.openfoam.org/download/
  - Current version 2.3.0 (as of Sept., 2014)
  - You can also get old versions. We only use version 2.2.2 since I haven't try the latest version.
  - Different flavors
    - Ubuntu Deb Pack
    - SuSE RPM Pack
    - RHEL Pack
    - Source Pack
    - Git Repository
- Install pre-compiled OpenFOAM® version 2.2.2
  - For example on an Ubuntu computer, go to:
    http://www.openfoam.org/archive/2.2.2/download/ubuntu.php
  - Follow the instructions on the webpage.
  - The default installation location is /opt, which requires you as root.
  - However, most time on a shared computer, you don't have the root right. Your best option is to install in your own directory from source (see next).

Installation of OpenFOAM®

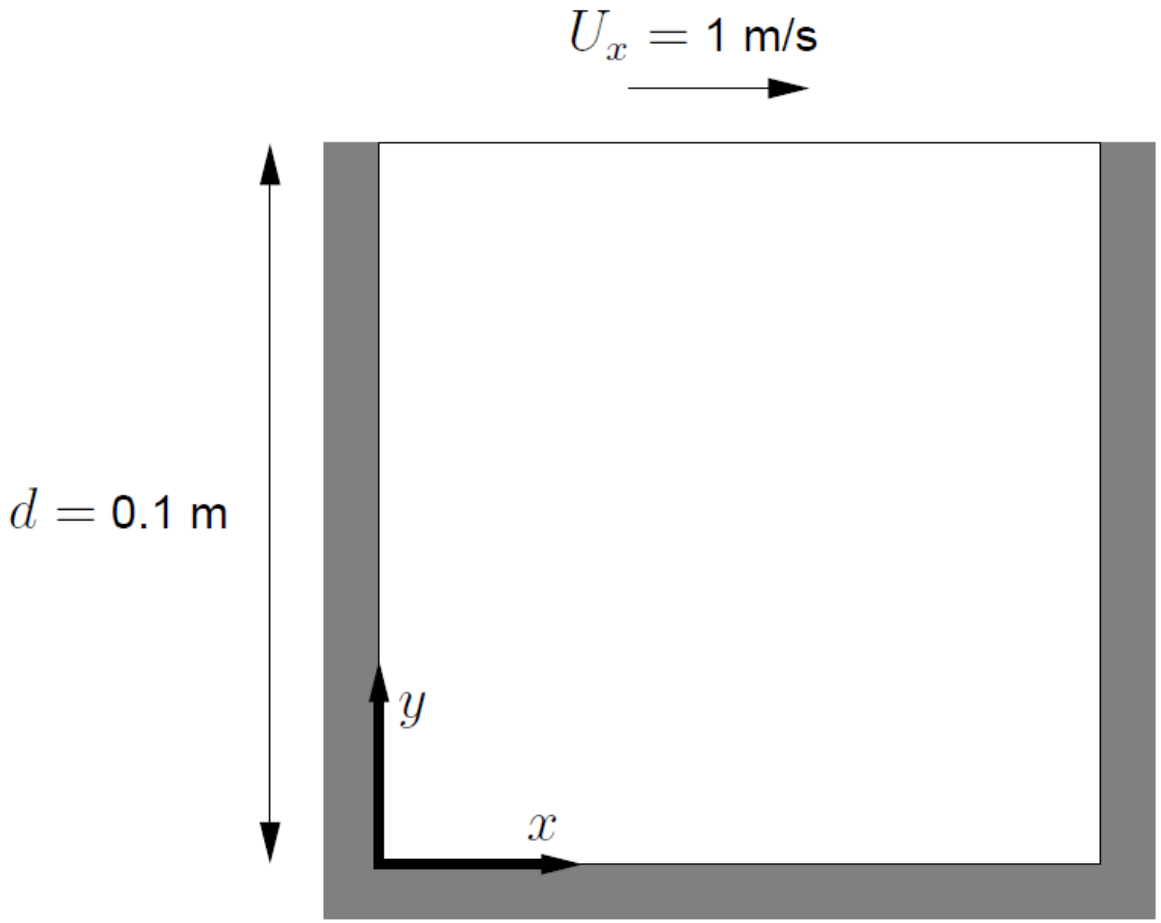
- Install OpenFOAM® version 2.2.2 from source
  - weblink: http://www.openfoam.org/archive/2.2.2/download/source.php
  - Follow the instructions on the webpage on how to compile from the source. Some suggestions:
    - If you have total control of the computer, such as your own personal computer, you can re-install the latest version of a Linux distribution, such as Ubuntu 14.04.1 LTS. This will make sure all the dependent libraries and gcc compiler are all updated.
    - If you don't have control of above, such as the supercomputers at PSU RCC, sometime you need to recompile some of the dependent libraries such as openmpi or even the gcc compiler (version $>$ 4.5.0) itself.
- You can install multiple versions of OpenFOAM® on your computer.
  - Every time just load the one you want.
  - The selection of current OpenFOAM® version is through Linux environmental variables.

Test the installation

- Walk around in the OF system
- Run first simulation

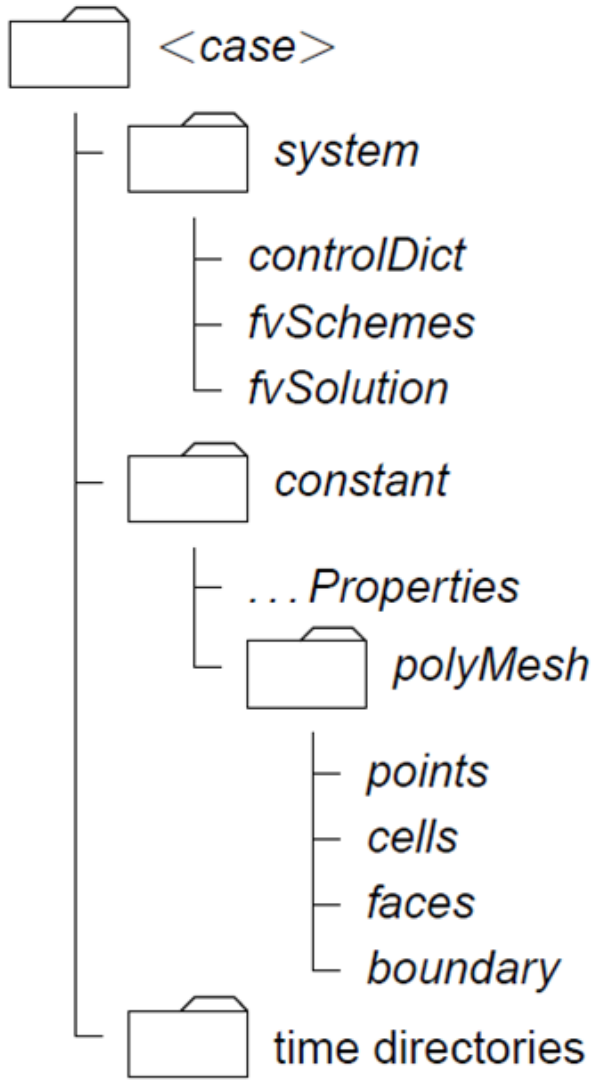
I will demonstrate these steps. You will have your chance in the exercises.

- The first simulation case: 2D lid-driven flow



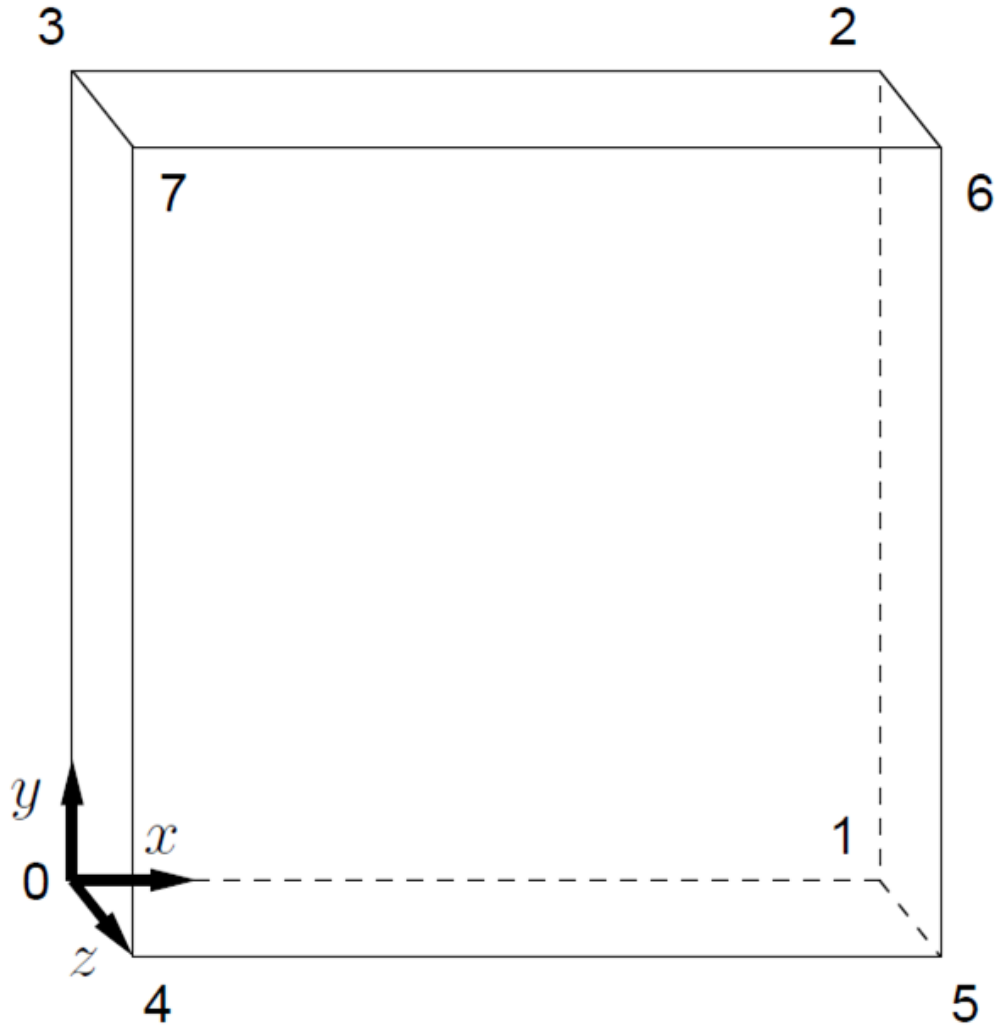
Test the installation

- Directory structure of an OpenFOAM® simulation case



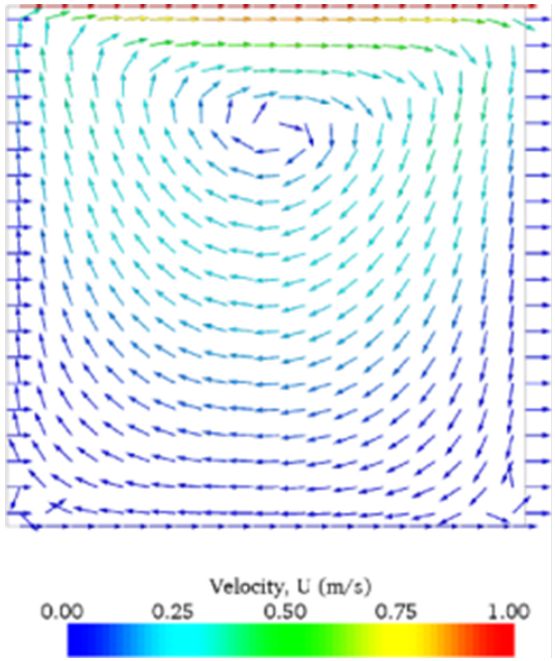
Test the installation

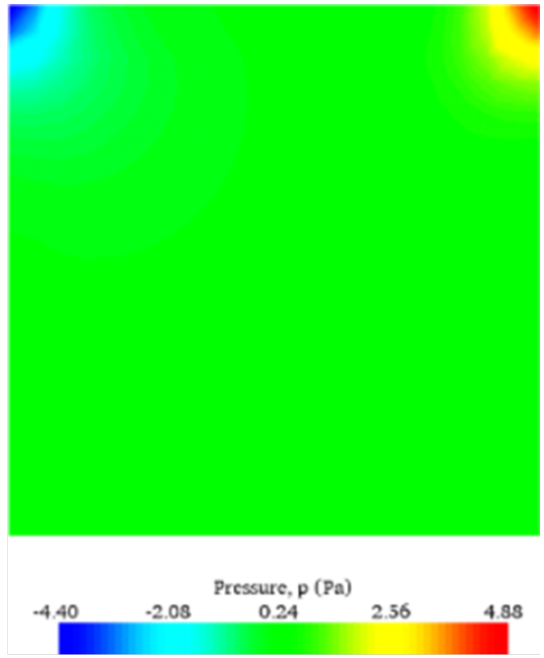
- Definition of the geometry for `blockMesh`



Test the installation

- Simulation results





Initial browsing of the OpenFOAM® code

We will very briefly browse the code structure of OpenFOAM® .

As our course progresses, you should get more and more familiar with the code structure. This familiarity helps you be an efficient OpenFOAM® code developer and user.

Source of information

To get more information:

- OpenFOAM® official web site: http://www.openfoam.org
- Online OpenFOAM® C++ source code documentation: http://www.openfoam.org/docs/cpp/
- OpenFOAM® discussion board: http://openfoam.cfd-online.com/cgi-bin/forum/discus.cgi
  Register and get the latest updates
- OpenFOAM® wiki: http://openfoamwiki.net/index.php/Main_Page
- User's guide and Programmer's guide
- And internet search engines: Google, yahoo, bing, etc.