

# OpenFOAM on Your Own Laptop

The teaching is designed so that you should follow the steps I do on a computer. This means that it is highly recommended that you bring your own laptop prepared for one alternative of running OpenFOAM. Note that there is no time to do any of this when the course has started, so everything must be done before the start of the course. The Linux commands below are written for bash shell. The only difference to make it work in tcsh shell is to everywhere change “2>&1 | tee” to “|& tee” (this means that standard output and standard error, i.e. the text that usually ends up in the terminal window, are sent both to the terminal window and the file specified after “tee”. The terminal window then still works as usual, but you can also go back and check if everything went well, by examining that log file.)

By following the instructions below, you should be able to install Ubuntu 10.04 LTS, OpenFOAM-2.0.x, OpenFOAM-1.6-ext (one optimal and one debug version), PyFoam, Python packages, Gnuplot etc., and later perhaps some additional useful things that we will use in the course.

Note that I am (unfortunately) using Word to generate this web page, which means that Word sometimes decides to change some characters to other characters that look the same, but aren't (e.g. – is not the same as - or -- and “ is not the same as "). I think that I have fixed this everywhere, but you never know. That is one reason for copy-pasting single rows rather than multiple rows, and check that all went well. Another reason is that some lines will require you to press a key at some point, such as the “sudo apt-get” lines. Let me know if you run into problems, and I'll help you.

## 1. Install and update Ubuntu 10.04 LTS

I have chosen to stick with 10.04 LTS since it is a Long-Term Service version that is supported until 2013. Hopefully we will then avoid some problems with cutting-edge versions. Note that you can easily install Ubuntu side-by-side with your Windows installation, and choose at start-up which operating system to use.

Follow the instructions at <http://www.ubuntu.com>, specifically <http://www.ubuntu.com/download/ubuntu/download>. Choose download option Ubuntu 10.04 LTS – Long term support, and 32 bit (recommended although the descriptions below have also been verified on a 64 bit laptop with Ubuntu 10.04 LTS in 64 bit mode. The OpenFOAM installation scripts automatically recognize whether it is a 32 or 64 bit system.).

You can optionally use a Windows installer: <http://www.ubuntu.com/download/ubuntu/windows-installer>

Start your computer using Ubuntu 10.04 LTS

Set up an Internet connection (System/Preferences/NetworkConnections). You will need an Internet connection during most of the compilation process of both Ubuntu and OpenFOAM, since many packages will be downloaded. Update the system (System/Administration/UpdateManager -> InstallUpdates)

Restart

Check the UpdateManager to make sure that all packages have been updated, or repeat the update procedure.

## 2. Install OpenFOAM

We will install OpenFOAM-2.0.x and OpenFOAM-1.6-ext. We will choose to pull the git source and compile from scratch. Both versions can also be installed as packages, which is much faster, but gives less control in my opinion (<http://openfoam.com/download/ubuntu.php>, and [http://openfoamwiki.net/index.php/Installation#OpenFOAM-dev\\_subversion\\_repository](http://openfoamwiki.net/index.php/Installation#OpenFOAM-dev_subversion_repository) – section 2.3.2). Feel free to try it out if you like – you can have both those versions and the git versions installed at the same time!

The installation procedure below has been tested on a freshly installed and updated Ubuntu 10.04 LTS. It should (hopefully) work on more recent versions as well (except perhaps for the specific tweaks that are related to Ubuntu 10.04 LTS). Note that the packages installed for OpenFOAM-2.0.x are needed also for OpenFOAM-1.6.x, in case you want to install only that version, therefore those packages are installed in this general section. The course is mainly based on 1.6-ext, since it has features that we will use in the course, that are not available in OpenFOAM-2.0.x. Discussions on differences between the versions can be included in your project report.

General preparations:

Applications/Accessories/Terminal - right-click and add to panel

Open a terminal window by clicking on the icon with a monitor in the top panel (which you just added to the panel), and copy-paste the following into it:

```
mkdir $HOME/OpenFOAM
sudo apt-get install git-core #(we will use git to get both versions of OpenFOAM)
sudo apt-get install subversion #(we will use svn to get some material during the course)
sudo apt-get install build-essential flex cmake zlib1g-dev qt4-dev-tools libqt4-dev gnuplot libreadline-dev libxt-dev
#Required by OpenFOAM-2.0.x
sudo apt-get install binutils-dev #Needed for -liberty in OpenFOAM-1.6-ext, but not needed for OpenFOAM-2.0.x
```

Now you can choose to install OpenFOAM-2.0.x and/or OpenFOAM-1.6-ext following the instructions below. While writing the instructions, I first installed OpenFOAM-2.0.x, but I see no reason why the instructions for OpenFOAM-1.6-ext should not work directly (let me know if you run into problems).

## 2.1 Install OpenFOAM-2.0.x:

We here more or less follow the instructions at [www.openfoam.com](http://www.openfoam.com), but with a new cmake installation.

Copy/paste into a terminal window:

```
cd $HOME/OpenFOAM
git clone git://github.com/OpenFOAM/OpenFOAM-2.0.x.git
wget --no-check-certificate http://downloads.sourceforge.net/project/foam/foam/2.0.0/ThirdParty-2.0.0.gtgz?r=http%3A%2F%2Fopenfoam.com%2Fdownload%2Fgit.php&ts=1312377741&use_mirror=netcologne
(note that you should now have a file named ThirdParty-2.0.0.gtgz – it might be that you have a filename including everything at the end of the above expression, then rename that file to ThirdParty-2.0.0.gtgz before proceeding)
tar xzf ThirdParty-2.0.0.gtgz
rm ThirdParty-2.0.0.gtgz
mv ThirdParty-2.0.0 ThirdParty-2.0.x
```

Update OpenFOAM-2.0.x later on, every now and then, by copy-pasting the following into a terminal window:

```
cd $HOME/OpenFOAM/OpenFOAM-2.0.x
git pull
```

Note that if you did your clone of OpenFOAM-2.0.x prior to 15<sup>th</sup> August 2011, you first need to change to url = git://github.com/OpenFOAM/OpenFOAM-2.0.x.git in the file OpenFOAM-2.0.x/.git/config, see [notes about the new OpenFOAM Foundation](#).

Create an alias (OF20x) to set up the environment by copy-pasting the following into a terminal window. Here we also copy the etc/bashrc file to etc/myBashrc since we want to modify it without touching the original file. We do the same thing for a bashrc file that is related to Paraview, and we will later do a modification to it.

```
echo "alias OF20x='source $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/myBashrc'" >> $HOME/.bashrc
cp $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/bashrc $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/myBashrc
sed -i s/"paraview.sh"/"myParaview.sh"/g $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/myBashrc
cp $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/config/paraview.sh $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/config/myParaview.sh
```

Close the terminal window and open a new one. The alias OF20x is from here on used to manually set up the OpenFOAM-2.0.x environment in each terminal window you open. We use an alias since we will have at least two different installations of OpenFOAM, and we want to be sure which one we are using.

Compile OpenFOAM-2.0.x by copy-pasting the following into the terminal window and waiting many hours (can be halted with CTRL-C and continued by the same commands):

```
OF20x
foam
./Allwmake 2>&1 | tee log_Allwmake_0
```

When done, check log\_Allwmake\_0 for error messages. Contact me if you have some.

Now OpenFOAM-2.0.x is compiled and ready to be used in each terminal window where you type OF20x.

The following is to make the post-processor Paraview work.

Now we compile Paraview 3.10.1 and the PV3FoamReader Module

This requires Qt version 3.6.2 or newer and cmake version 2.8.2 or newer

Check by copy-pasting into a terminal window:

```
qmake -v
```

```
cmake --version
```

In Ubuntu 10.04 LTS, Qt is ok (4.6.2), but cmake is not ok (2.8.0) and this is why we modified the bashrc files previously.

Cmake can be found at <http://www.cmake.org>. Install the latest release by copy-pasting into a terminal window (two options):

- General compilation of cmake-2.8.5, using the makeCmake script  
First make OpenFOAM aware of which cmake version to compile and use  
sed -i s/"cmake-2.8.3"/"cmake-2.8.5"/g \$WM\_THIRD\_PARTY\_DIR/makeCmake  
sed -i s/"cmake-2.8.4"/"cmake-2.8.5"/g \$HOME/OpenFOAM/OpenFOAM-2.0.x/etc/config/myParaview.sh  
Open a new terminal window and continue with downloading and compiling  
OF20x  
cd \$WM\_THIRD\_PARTY\_DIR  
wget http://www.cmake.org/files/v2.8/cmake-2.8.5.tar.gz  
tar xzf cmake-2.8.5.tar.gz  
rm cmake-2.8.5.tar.gz  
./makeCmake
- Quick binary release installation for 32 bit only (not recommended)  
sed -i s/"cmake-2.8.1"/"cmake-2.8.5-Linux-i386"/g \$HOME/OpenFOAM/OpenFOAM-2.0.x/etc/config/myParaview.sh  
cd \$HOME/OpenFOAM/ThirdParty-2.0.x/platforms/linuxGcc  
wget http://www.cmake.org/files/v2.8/cmake-2.8.5-Linux-i386.sh  
chmod +x cmake-2.8.5-Linux-i386.sh  
./cmake-2.8.5-Linux-i386.sh  
Press q to skip all the intro text  
Press y (and enter) to accept the licence  
Press Y (and enter) to accept installing in default subdirectory name (cmake..)

Now we will compile Paraview and the readers by copy-pasting into the terminal window:

```
cd $WM_THIRD_PARTY_DIR
```

```
./makeParaView 2>&1 | tee log_makeParaView
```

```
cd $FOAM_UTILITIES/postProcessing/graphics/PV3Readers
```

```
wmSET
```

```
./Allwclean
```

```
./Allwmake 2>&1 | tee log_Allwmake
```

Now OpenFOAM-2.0.x is compiled and ready to be used in each terminal window where you type OF20x

Time to test by copy-pasting into the terminal window:

```
mkdir -p $FOAM_RUN
```

```
run
```

```
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity .
```

```
cd cavity
```

```
blockMesh
```

```
checkMesh
```

```
icoFoam
```

```
paraFoam
```

Close the terminal window if you will continue with the installation of OpenFOAM-1.6-ext, so that there are no environment conflicts!

## 2.2 Install OpenFOAM-1.6-ext (with optional hotfix/ThirdParty):

First a useful link, added after the course 2011: <http://www.cfd-online.com/Forums/openfoam-installation/94137-openfoam-1-6-ext-git-installation-ubuntu-11-10-x64-3.html#post339219>

At the time of writing, the ThirdParty directory of OpenFOAM-1.6-ext had bugs, so here is a slightly more complicated description where we merge those bug fixes. At the same time we will learn some git.

Open a fresh terminal window.

Pull the source code using git, by copy-pasting into the terminal window:

```
cd $HOME/OpenFOAM
```

```
git clone git://openfoam-extend.git.sourceforge.net/gitroot/openfoam-extend/OpenFOAM-1.6-ext
```

Update OpenFOAM-1.6-ext later on, every now and then, by copy-pasting the following into a terminal window:

```
cd $HOME/OpenFOAM/OpenFOAM-1.6-ext
```

```
git pull
```

Let us examine an option with git by making a branch by copy-pasting into a terminal window:

```
cd $HOME/OpenFOAM/OpenFOAM-1.6-ext
```

```
git checkout -b myBranch #Creates a branch called myBranch, and checks it out
```

By creating a new git branch, we can actually modify the files in myBranch without modifying the files in the master branch. At this stage you don't have to bother about that. You just have to make sure that you are using myBranch, by copy-pasting into a terminal window:

```
cd $HOME/OpenFOAM/OpenFOAM-1.6-ext
```

```
git branch -a
```

There should be a \* in front of myBranch, meaning that we will work in the myBranch branch.

The branches remotes/\* are developments that have not been merged into the master branch.

At the time of writing these instructions, there were bugs in the ThirdParty scripts of the master branch.

Those were fixed in a branch called remotes/origin/hotfix/ThirdParty\_scripts, so I had to merge those fixes by copy-pasting the following into the terminal window (if that branch is not there anymore, you shouldn't do this!):

```
git merge remotes/origin/hotfix/ThirdParty_scripts
```

There was also some work being done on documenting the ThirdParty installation procedure, so to get that new information, copy-paste the following into the terminal window (if that branch is not there anymore, you shouldn't do this!):

```
git merge remotes/origin/hotfix/ThirdParty_documentation
```

Create the alias OF16ext by copy-pasting into a terminal window:

```
echo "alias OF16ext='source $HOME/OpenFOAM/OpenFOAM-1.6-ext/etc/bashrc'" >> $HOME/.bashrc
```

Close the terminal window, open a new one, and type

```
OF16ext
```

Before compiling the ThirdParty packages, some notes should be made on the compilation process of OpenFOAM-1.6-ext. In \$FOAM\_INST\_DIR/OpenFOAM-1.6-ext/etc there are two files, prefs.sh-EXAMPLE and prefs.csh-EXAMPLE. In those you can see some examples on how to control the choice of system/ThirdParty-installed ThirdParty products. Copy the file corresponding to the shell you are using (bash by default in Ubuntu 10.04 LTS, so the .sh file) to a new file with the same name without "-EXAMPLE"

(e.g.: cp prefs.sh-EXAMPLE prefs.sh)

```
cd $FOAM_INST_DIR/OpenFOAM-1.6-ext/etc
```

```
cp prefs.sh-EXAMPLE prefs.sh
```

By default, it is specified that the system Qt should be used, but for that we need to specify the PATH to Qt. Optionally, we can choose to install Qt in our ThirdParty directory, which gives us a Qt installation specific to the OpenFOAM installation, but requires some more hours of compilation and some more disk usage. In any case, we need to do something to make the compilation work for the post processor. Here, both choices are shown

- Use the system Qt (recommended in Ubuntu 10.04 LTS)
 

```
sed -i s/"#export QT_DIR=path_to_system_installed_qt"/"export QT_DIR=/usr/share/qt4"/g
prefs.sh
sed -i s/"#export QT_BIN_DIR"/"export QT_BIN_DIR"/g prefs.sh
```
- Install and use your own ThirdParty version of Qt (not recommended in Ubuntu 10.04 LTS):
 

```
sed -i s/"#export QT_THIRD_PARTY=1"/"export QT_THIRD_PARTY=1"/g prefs.sh
```

I also prefer not to have a lot of lines written to the terminal window when I set up the OpenFOAM

environment, so then I do

```
sed -i s/"export FOAM_VERBOSE=1"/"#export FOAM_VERBOSE=1"/g prefs.sh
```

During the compilation process of the ThirdParty products the system needs to have gmake and libstdc++.so.5, otherwise it will complain and not compile (“/bin/sh: gmake: not found” and “error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory”). This can be fixed in Ubuntu 10.04 LTS by copy-pasting into a terminal window (gmake and make are the same, just different names):

```
sudo ln -s /usr/bin/make /usr/bin/gmake
```

Then go here:

For 32-bit: <http://packages.debian.org/lenny/i386/libstdc++5/download>

For 64-bit: <http://packages.debian.org/lenny/amd64/libstdc++5/download>

Click on one of the mirror links (e.g. ftp.de.debian.org/debian)

Choose "Open with GDebi Package Installer (default)" and install the package

We also need to install two packages, by copy-pasting into a terminal window:

```
sudo apt-get install libxext-dev libxrender-dev
```

Close the terminal window, open a new one and type:

```
OF16ext
```

Start compiling the ThirdParty packages by copy-pasting into the terminal window and waiting many hours (NOTE that an internet connection is required, since source code will be fetched. The compilation process can be halted with CTRL-C and continued by the same commands, although the interrupted package will be compiled from scratch again. Stage 4, Qt and Paraview are the packages taking most of the time):

```
sudo apt-get install rpm #The ThirdParty compilation process of 1.6-ext is based on rpm
```

```
cd $WM_THIRD_PARTY_DIR
```

```
./AllMake 2>&1 | tee log_AllMake
```

This should now have been installed (in \$WM\_THIRD\_PARTY\_DIR/packages, qt-everywhere-opensource-src-4.7.0 only if you specified above that you wanted your own installation of Qt):

```
cmake-2.8.3    openmpi-1.4.3    qt-everywhere-opensource-src-4.7.0
```

```
libccmio-2.6.1  ParaView-3.8.1    scotch-5.1.10b
```

```
mesquite-2.1.2  ParMetis-3.1.1
```

```
metis-5.0pre2  ParMGridGen-1.0
```

Compile OpenFOAM-1.6-ext by closing the terminal window, opening a new one, copy-pasting into a terminal window and waiting many hours (can be halted with CTRL-C and continued by the same commands):

```
OF16ext
```

```
foam
```

```
./Allwmake 2>&1 | tee log_Allwmake_1
```

When I did this I got some problems like:

```
/usr/bin/ld: cannot find -lfaceDecompositionMotionSolver
```

The reason for this is just that the compilation order is not correct.

(Bug-reported, see <http://sourceforge.net/apps/mantisbt/openfoam-extend/view.php?id=76>, perhaps it has been fixed)

Just run the link process again, and hopefully all will go through:

```
./Allwmake 2>&1 log_Allwmake_2
```

The Paraview readers are compiled separately, by copy-pasting into the terminal window:

```
cd $FOAM_UTILITIES/postProcessing/graphics/PV3FoamReader
```

```
./Allwclean
```

```
./Allwmake 2>&1 | tee log_Allwmake
```

Now OpenFOAM-1.6-ext is compiled and ready to be used in each terminal window where you type OF16ext

Time to test by copy-pasting into the terminal window where you already typed OF16ext:

```
mkdir -p $FOAM_RUN
```

```
run
```

```
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity .
```

```
cd cavity
```

```
blockMesh
checkMesh
icoFoam
paraFoam
```

In the course you will learn how to use a debugger with OpenFOAM-1.6-ext, and for that we need to re-compile OpenFOAM-1.6-ext according to the following procedure:

Create a new alias by copy-pasting into a terminal window:

```
echo "alias OF16extDebug='export WM_COMPILE_OPTION=Debug; source \$HOME/OpenFOAM/
/OpenFOAM-1.6-ext/etc/bashrc'" >> \$HOME/.bashrc
```

Before proceeding, we need to do a fix to avoid having to re-compile the ThirdParty packages in Debug mode:

In \$FOAM\_INST\_DIR/OpenFOAM-1.6-ext/etc/settings.sh, before line saying "# convenience", add:

```
export WM_OPTIONS=$WM_ARCH$WM_COMPILER${WM_PRECISION_OPTION}Opt
```

At the end of the same file, add:

```
export
WM_OPTIONS=$WM_ARCH$WM_COMPILER$WM_PRECISION_OPTION$WM_COMPILE_OPTION
```

I.e. I temporarily hard-code that the Opt version of the ThirdParty packages should be used.

Open a new terminal window and type (again, wait many hours):

```
OF16extDebug
foam
./Allwmake 2>&1 | tee log_Allwmake_Debug_1
./Allwmake 2>&1 | tee log_Allwmake_Debug_2
cd $FOAM_UTILITIES/postProcessing/graphics/PV3FoamReader
./Allwclean
./Allwmake 2>&1 | tee log_AllwmakeDebug
Done!
```

Now you have two installations of OpenFOAM-1.6-ext, and you choose which one to use by opening a new terminal window and typing either OF16ext or OF16extDebug. The debug version is slower, and should only be used when debugging.

We will do debugging with gdb and Emacs. Emacs needs to be installed:

```
sudo apt-get install emacs23
```

You can also try ddd (not available at Chalmers):

```
sudo apt-get install ddd
```

Close the terminal window if you will continue with the installation of OpenFOAM-2.0.x, so that there are no environment conflicts!

### 3. Install PyFoam (and Python, Numpy, SciPy, Matplotlib, Gnuplot)

First, just to make sure that we have all the packages we will need later, by copy-pasting into a terminal window:

```
sudo apt-get install python python-dev python-matplotlib python-numpy python-scipy gnuplot
```

We will follow the installation instructions for PyFoam at [http://openfoamwiki.net/index.php/Contrib\\_PyFoam](http://openfoamwiki.net/index.php/Contrib_PyFoam)

Condensed installation instructions:

```
mkdir $HOME/OpenFOAM/linuxSrc
cd $HOME/OpenFOAM/linuxSrc
wget http://openfoamwiki.net/images/b/b8/PyFoam-0.5.6.tar.gz
tar xzf PyFoam-0.5.6.tar.gz
rm PyFoam-0.5.6.tar.gz
cd PyFoam-0.5.6
```

- Install as regular user (recommended):

```
python setup.py install --prefix=$HOME/OpenFOAM/PyFoam-0.5.6
echo "alias PF='export PYTHONPATH=$HOME/OpenFOAM/PyFoam-0.5.6/lib/python-2.6/site-
packages:$PYTHONPATH; export PATH=$HOME/OpenFOAM/PyFoam-0.5.6/bin:$PATH'" >>
```

```
$HOME/.bashrc
```

- Install as root (not recommended):  
sudo python setup.py install

Open a new window and type (<TAB> refers to the tab key):

PF #Only if installed as regular user

pyFoam<TAB><TAB>

You will see numerous pyFoam commands that can be used to help you use OpenFOAM efficiently.