# Spring 2017- ME614 - Homework 0

## Computing Environment Setup

(Due January 20, 2017)

Computational Fluid Dynamics (CFD) research is mainly carried out on supercomputers and these architectures only run Unix-based systems. It is therefore highly recommended that you develop, and run all your codes using a Unix-based environment. Many guides on Unix/Linux can be found online. It is recommended that you take the time to complete as many tutorials as you can within this set, after you gain access to a Unix/Linux machine. The following includes technical details that will allow you to set up your working environment.

## 1 Machine Setup

Your workflow will involve two machines: your personal machine and a Linux server. You will work on your homework assignments from your personal machine and upload them to the Linux server, from which the instructor will automatically download your work.

### 1.1 Your Machine

You can work with your machine of choice (your laptop or desktop, at home or at work) and with any operating system you like. It is however recommended that you choose a Unix-based working environment such as MacOSX or a distribution of Linux. Windows is NOT a Unix-based operating system.

#### 1.1.1 Linux/MacOSX

Most of the software that you will need already comes pre-installed on Unix/Linux[1] and MacOSX machines. It is strongly recommended that you install TeX Live to generate production quality figures.

#### 1.1.2 Windows

If you (unfortunately) decide to use Windows, there are several ways to obtain a Unix-like working environment. A non-invasive option is Cygwin, which will provide you with a Unix-like terminal from which you can access the content of your course folder (somewhere in `/cygdrive/c/`). Please select `ssh` and `git` during Cygwin's installation.

Alternatively, you can install Ubuntu (or any other distribution of Linux) in dual-boot mode or bootable from within Windows itself. If you do this, you should follow the instructions in section 1.1.1 and section 2.2 (as if you had an actual Linux machine).

[1] Linus Benedict Torvalds, born December 28, 1969, is a Finnish-American software engineer, who is the creator of the Linux kernel and for a long time, principal developer. Linus Torvalds had a "fixation for flightless, fat waterfowl" and Torvalds claims to have contracted "penguinitis" after being gently nibbled by a penguin: "Penguinitis makes you stay awake at nights just thinking about penguins and feeling great love towards them." Torvalds' supposed illness is a joke, but he really was bitten by a little penguin on a visit to the National Zoo & Aquarium, Canberra, Australia. Torvalds was looking for something fun and sympathetic to associate with Linux, and a slightly fat penguin sitting down after having had a great meal perfectly fit the bill.

### 1.2 Linux Server

Regardless of the coding language, personal machine or operating system of choice, you are required to configure a Linux server (that is, a second machine), to which you will upload your homework assignments. Two Linux machines are physically located in ME2028 with accounts on them (that should be) set up for you. Their network addresses are:

```
wind.ecn.purdue.edu, water.ecn.purdue.edu
```

If you are a non-ME student, you probably do not have an account on these machines. In that case, you can use any linux machine that is available to you that is accessible via `ssh`. For example, if you are in nuclear engineering, the following machine(s) may be available to you

```
helios.ecn.purdue.edu
```

If you are in aerospace engineering, the following machine(s) may be available

```
dragon.ecn.purdue.edu, thrust.ecn.purdue.edu, gus.ecn.purdue.edu
```

Remember to inform the instructor of the network address of your server.

You do not necessarily need to be on Purdue's network to log into your machine. Only in certain cases, due to IP restrictions, you might need a VPN to connect to these machines. You can log on to any of the aforementioned machines from your personal machine using the SSH protocol, by typing, from the terminal

```
ssh -XY <your-purdue-login>@<linux-machine-address>
```

If the ssh login does not accept your Purdue password, then your account is probably not set up. Please contact ITaP to fix the problem as soon as possible.

A list of all of the computing machines available to anybody in the department can be found here:

```
https://engineering.purdue.edu/MECL/Public/machine_loads/general.whtml
```

These machines will be useful for some parallel computing exercises that might be discussed later in the course.

## 2 Python Setup

It is highly recommended (but not mandatory) that you make Python your coding language of choice. Examples and support throughout the course will only be offered in Python version 2.x. Running the scripts provided in class with Python version 3.x will NOT work. If you are already familiar with Matlab, switching to Python will be easy and rewarding.

### 2.1 Windows

To install Python on Windows follow the tutorial that is provided to you. It is NOT recommended you install Python within Cygwin, which is simply meant to allow you to connect with a Linux machine via `git` and `ssh`.

### 2.2 Unix/Linux/MacOSX

The Anaconda distribution of Python is the recommended choice for either MacOSX or Linux operating systems. In particular, it is recommended to use Miniconda and progressively install the required packages via the command

```
conda install <package-name>
```

This allows the advanced installation of applications outside of the Anaconda distribution, such as hdf5-based parallel I/O, manual installation of openMPI, etc.

# 3 Git-Repository Setup

You need to set up your Linux machine of choice as a server to host your work. Log ON to your Linux Server and start by setting up the SSH environment with the following commands:

```
cd ~
mkdir .ssh && chmod 700 .ssh
touch ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys
```

Generate your SSH key on the Linux server by typing

```
ssh-keygen -t rsa -C "<your-purdue-email-address>"
```

and then press enter three times (no passphrase). This will create the folder `~/.ssh` with your public and private SSH key in it ON the Linux server. Log OUT of the Linux server and REPEAT these steps in the Linux environment that you have set up on YOUR personal machine (section 1.1).

In order to grant the instructor and the TA password-free access to your Linux server, log on to the Linux server and type:

```
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDV57TbLDCmOi74durfpXlKuNBSIMRldvnyuER2wZN2cTZdkcDogiL378sZq
    ↪ wBGDMW4N7NbvYnjEwnp38y2bgKStjb8C4Hzur+l9P1+m6z6QXRKlZIMKYnM1nuYbgFoxkWcbudQOOesiXf8P4yPOmYLhXyEO
    ↪ Le7xZJg5xKrBN2nS8YIrnbcZzu9J8SAsEo3wT9KeGGODAvKfu7PRWNHiO4eLwnDYgMG4DnXexNS8WQ5bRaWCm1t9wMhyLFkl
    ↪ luePNE4rv8zxApXOamsR5ws69fRCAWP1TVMb4+/AxItFb+yLutnBJiTPzkaKzrPaOE83PBQLB47Tlfj+jl8+FPH/EJb scal
    ↪ o@purdue.edu" >> ~/.ssh/authorized_keys
```

and

```
echo "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA3zREgzaR6md8M2LetKNGOMeoH2MDH4zw6wUc4ybtiwCz4JpSUbw+duXHWyxBh
    ↪ A7bQ1ANefuDcwwTSVG1EL6yz1CDQR/pBhETlqayri5CCitIBy4RthpYrMhX36LsCAy6TEfWmcC6dfHDbS6OIryDw7QrEOpZj
    ↪ slrOYlBOIlkpYMGy9GknxPY5esD+pFa4GtcHYujXoo5rz1sP+XyC7Jkb4VJrW38YH+i1XgPl+REjGbo1VKLNuw23FKEPbpDj
    ↪ GOQKo93lOoOO+vrJF3dgbvMFFtRs/VgsoQqb8ZLCv+YOVe4hes7/62588K+4hHTpI1OI6OQHljGtbwLSY1oYcWFpw== pate
    ↪ l472@purdue.edu" >> ~/.ssh/authorized_keys
```

and

```
echo "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAw7H8E5wiHM9G+9zjGfjhspCfHdmyP73e3cELGqkTn5GZJsTpnhvpAwsbdF1HE
    ↪ tjNYT78IX73ey67VXHPSnHRtERB+F4AtgZFlc8aEXtcliEvIHpfYFTTk6IQ8kZDf0YErotW5n4+tXaeROIlRWA8lLeuCLMPu
    ↪ Hb6Nh80IRd+UU+V9Lk3yfW/7eCiK6VBge7lYSUvGF6jkPjZUryWl91X9CP+oRoJIOIsy287W2zZcd7ksJJb/NenAPlX+KJ43
    ↪ YwVb+DfNOw7F07efb6VUqjPlo2kVDThllFvS+Ho+ISO8ZlDbDCTg1NOyoPy2Nzp49bp2MZWsl4JkBi4+uVBl6EOOw== ki
    ↪ m1625@purdue.edu" >> ~/.ssh/authorized_keys
```

Extreme care should be taken in copying and pasting the command above in the terminal window: the command should be assigned with *one contiguous string of characters*, without missing any character, with the exception of the four red arrows you see, which should not be copied.

To grant yourself (i.e. your personal machine) password-free access to the Linux server, repeat the same step by substituting the long string within quotes above with the content of the file `~/.ssh/id_rsa.pub` that should now be present on YOUR personal machine. You should now be able to log on to your Linux server via SSH, as you did in section 1.2, without providing your password (don't worry, this is still safe).

You will use git to create a software repository on your Linux server. Every Unix-based systems nowadays comes with git preinstalled. To set up your git repository, on the Linux server type

```
mkdir -p ~/git-repositories/<semester><year>-mexxx
cd ~/git-repositories/<semester><year>-mexxx
git init --bare
```

where `<semester>` is the current semester (`spring`, `summer` or `fall`), `<year>` is the current year in a four digit format (`YYYY`) and `mexxx` indicates the current course number (`me608` or `me614`). The git repository folder name needs to be all in lower case.

Now, via a Linux terminal on YOUR personal computer navigate to your local MEXXX folder (for example, in your PhD_Purdue/Documents/Courses/me-xxx/folder )

```
cd <your-local-homework-folder>
```

and populate the git repository

```
git init
touch test_file
git add .
git commit -m "initial commit"
git remote add origin <purdue-login>@<linux-machine-address>:~/git-repositories/<semester><year>-mexxx
git push origin master
git push --set-upstream origin master
```

## 4 Homework Submission

On your personal machine, organize homework folders in the following way

```
<your-local-homework-folder>/homework0/report/
<your-local-homework-folder>/homework0/code/
```

in the `/code/` folder insert the Python script `homework.py` that will be provided to you for this homework. Running it will generate a PDF figure in the `/report/` folder plotting a cosine and sine function. It is required you organize all of your future homework folders EXACTLY in this way, regardless of your coding language of choice. To execute the script provided a full installation of `texlive` may be required.[2]. If you do not want to work with LaTeX, then you can run the other script provided: `homework_noLaTeX.py`. Successfully running either one of the scripts is enough to complete homework zero. If Python is not your language of choice, you are required to upload any code that will generate a production-quality plot of a cosine and a sine wave as provided to you.

When you are done, you need to upload your work to the Linux server, by typing from your local machine:

```
cd <your-local-homework-folder>
git add .
git commit -m "final submission: homework zero"
git push
```

The instructor should now be able to automatically download your homework from the server. You can upload your work (add/commit/push) on the Linux server any time and as many times as you want before the due date.

**Note**: Do not panic if you are unable to view your uploaded homework files while navigating through your git repository on your linux server machine. This is a feature of “bare” repositories and should not be mistaken for an issue with your submission. If you wish to be notified every time your repository is changed, look up how to set up git e-mail notifications. Add yourself, but do not add the instructor or the TA’s to this notification system.

[2] On Ubuntu this is simply done by typing: `sudo apt-get install texlive-full` and/or `sudo apt-get install texlive-latex-extra`