

0-Preliminaries

March 16, 2022

1 Preliminaries

1.1 Course Philosophy

This class will follow a so-called “flipped” approach where most of the course material is given in advance of the lectures, and it is the students’ responsibility to come prepared having reviewed the material. The lecture time is then used for higher-level learning activities, including deeper discussion of the course content and implementing the concepts into computer codes. Therefore, it is essential that students come to class having fully reviewed the relevant material, and that they bring a laptop with them to work on problems in class.

1.2 Python Programming

This course requires some knowledge of the [Python](#) programming language, although this skill will be developed naturally as you progress through the course. It is recommended that you do attain at least a basic knowledge of Python prior to moving on to the next lesson.

We will be using Python 3 for this course, which is not backwards compatible with Python 2. Be sure that you have Python installed on your computer before continuing. The [Anaconda version of Python](#) is recommended, since it comes with many of the required packages for this course.

Exercise: Install Python 3 on your computer.

Exercise: Complete the [Plotting and Programming in Python](#) workshop from the [Software Carpentry Foundation](#).

1.3 Jupyter Notebooks

This course will be conducted using [Jupyter Notebooks](#), such as the one you are reading now, as the main method for delivering content and for submitting assignments. If you installed Anaconda above, then you already have Jupyter installed. If you installed another version of Python, then you should install Jupyter now.

Exercise: Ensure that you have Jupyter installed on your computer. [Click here](#) for installation information.

1.3.1 Notation Conventions

You have already been given a few exercises to complete. Here are the remaining alert boxes that will be used in this course for different purposes.

Tip: These boxes will be used to offer tips to help you work through the material more efficiently.

Warning: These boxes will be used to highlight common errors or pitfalls.

Exercise: These boxes will be used to assign exercises that recommended to be completed before moving on in the course.

1.4 Git - Distributed Version Control

This course is not just intended to teach CFD, but it is also going to outline some important best practices for developing engineering software. The first step to developing better software is to keep a record of your changes and to allow for other people to use and contribute to your code. This is most readily accomplished using a distributed version control system, the most popular of which is [Git](#). Coupled with the local usage of Git, is the use of remote repositories which allows you to work on your code on multiple machines and collaborate with others. Some popular repositories are [Github](#), [BitBucket](#), and [GitLab](#).

You will be required to use Git to version control your assignments. You will hand them in by sharing access to the remote repository. Your commit history will be checked and will be a component of your grade.

Exercise: Install Git on your computer and become familiar with its basic operations. Read chapters 1 and 2 of the [Git Book](#) for installation information and an outline of the most basic commands.

2 Next Steps

Once you have set up Jupyter on your personal computer and have attained a basic knowledge of Python and Git, it is time to move on to the lesson on [Introduction to the Finite-Volume Method](#)

[]: