Math 246: Mathematical Computing and Programming Laboratory

Dr. Brenton LeMesurier

Fall 2015

Classes meet: Thursday, 1:40-4:10 pm in Maybank 200

Professor: Dr. Brenton LeMesurier
Office: Robert Scott Small Building, room 200
Phone: 953-5917, messages 953-5730 (but email is better for messages)
Email: lemesurierb@cofc.edu
Web Site: `http://blogs.cofc.edu/lemesurierb/`
Office Hours: To be arranged.

There is a site for this course in the College's Learning Management System OAKS at `http://lms.cofc.edu` and also accessible from MyCharleston, `http://my.cofc.edu`. This will be used for you to submit your programming work, and for me to respond with comments, suggestions for revisions, and grading.

Course Objectives and Student Learning Outcomes

The main expectation of this course is that students learn how to use computational software (such as a Python programing environment) to implement algorithms for computing numerical solutions to mathematical problems, and to present results appropriately with graphs, tables and such. This will usually be done in conjunction with taking the course MATH245, where algorithms for such problems are learnt.

Computers and Software

We will use the programming language Python 3 together with some add-on packages for scientific computing, like Numpy and SciPy for numerical computing tools, and Matplotlib for graphics. No familiarity with this software or computer programming is assumed; learning all that is what this course is for!

Aside: students who are very experienced with other computational software such as Matlab or Mathematica might be able use that instead for most of the work in this course and in Math 245: if you are considering that, discuss it with me.

This collection of software is most easily accessed through Anaconda, which is available in the computer classroom Maybank 200, and can be easily downloaded and installed on any Windows, Mac OS, or Linux computer. Within Anaconda, the main component that we use is the integrated development environment Spyder, which can also be installed and used as a stand-alone application. This stand-alone version is not necessary, but some of you might find it convenient.

The other part of Anaconda that we use is the *iPython notebook* system, now also called *Jupyter*. I will use this for many presentations, and you might want to use it to produce project reports in MATH 245.

You can get this software from the following links; be sure to get the version with Python 3.4, not Python 2.7.

Anaconda `hhttp://continuum.io/downloads#py34`

Spyder for Windows `https://bitbucket.org/spyder-ide/spyderlib/downloads/spyder-2.3.5.2.win-amd64-py3.4.exe`

Spyder for Mac OS X `https://bitbucket.org/spyder-ide/spyderlib/downloads/spyder-2.3.5.2-py3.4.dmg`

Spyder for Linux `https://bitbucket.org/spyder-ide/spyderlib/downloads/spyder-2.3.5.2.zip`

Coursework and Assessment

The work for this course will consist of about ten modules, with an initial version of the work for a module submitted one week, and often revisions done later, on the basis of my feedback.

Grading is based on successful completion after possible revisions, not partial credit for submissions that are incomplete or do not give correct results. Thus the grading scheme for each module will be:

4: Complete and correct in all respects.

3: Mostly complete and correct, but with minor omissions or room for refinement: can be revised to get to an A.

2: This might be given if one sub-task is successfully completed but another is still in progress.

Incomplete/ungraded: Feedback comments only: must be revised or completed to get to a grade.

I will give a lot of feedback and allow several revisions: with programming and most mathematics, working and checking until you get things completely right makes a lot more sense than partial credit for results that are partially wrong!

Before leaving the class each week, you should discuss your work with me, even if you know that it is unfinished. Then most work will be submitted to a dropbox in OAKS, though some modules will involve written work. I will respond through OAKS with comments, and either suggestions for revisions or a final grade.

Final Grading

The module grades will be averaged, and converted to a final letter grade for the course on the scale:
$[3.8, 4]$ for an A,
$[3.5, 3.8)$ for an A-,
$[3.2, 3.5)$ for a B+,
$[2.8, 3.2)$ for a B,
$[2.5, 2.8)$ for a B-,
$[2.2, 2.5)$ for a C+,
$[1.8, 2.2)$ for a C,
etc.

It is likely that you will finish most of the work for this course before the last weeks of class: the remaining meeting time will be available for supervised work on programming projects for MATH245, and/or for revising earlier modules to improve your grades.