PSCB57 – Introduction to Scientific Computing

Course Information

Lecture Times: Tues 12:00-13:00, Thu 15:00-16:00
Lecture Location: HW 214

Tutorial Times: Thu 10:00-11:00 (TUT0001), Thu 11:00-12:00 (TUT0002), Thu 14:00-15:00 (TUT0003)
Tutorial Locations: BV 498

Instructor: Gyula Lorincz
Office: S503C
Office Hours- Tue 14:00-15:00, Thu 16:00-17:00
Telephone: 416-287-7248

Co-Requisites: MATA36H or MATA37H or MATA35H (with permission of the instructor) and one A-level science course.

Course Description: An introduction to the use of computers in the physical and biological sciences. Choice and design of algorithms and their implementation in a high-level computer language for the solution of problems arising in the physical and biological sciences. Topics will include elementary numerical analysis such as numerical integration, mathematical modeling of physical systems, data fitting and interpolation. (Intended primarily for physical and biological science students who do not plan to pursue any of the Programs in computer science or cognitive science.)

Numerical Analysis is the mathematics of approximation, and is fundamental to quantitative science. A tentative list of numerical techniques to be discussed.

- Taylor Series
- Root Find
- Integration
- Differential Equations
- Data Modeling

Despite the approximate nature of these techniques, they are based on (exact) elementary theorems of calculus in particular, theorems relation to Taylor series with their convergence are of central importance. Proofs of relevant theorems will be discussed in lecture, but will not be rigorously developed. Students are expected to be able to fill in the details where required.

Practical application of numerical analysis requires implementing these techniques as computed algorithms. Algorithms will be implemented in the Python programming language. Python is a powerful and popular high-level programming language which supports the rapid development of numerical algorithms. Furthermore, Python contains most of the features of more powerful programming languages. Algorithms developed in Python are therefore easily portable to other programming environments.

The first part of the course will be devoted to developing a working knowledge of Python, no prior programming experience is required.

Textbook: There is no required textbook for the course. All necessary material will be presented in lectures and tutorials, or made available online. The following are some of the online references you may use:

Title & Author & Webpage

An Elementary Introduction to Scientific Computing
Charles C. Dyer, Peter S. S. Ip
http://pathfinder.utsc.utoronto.ca/~dyer/csca57/book_P/book.html

How to Think Like a Computer Scientist
Brad Miller, David Ranum
http://interactivepython.org/courselib/default/user/login?_next=/courselib/default/index

Dive Into Python
Mark Pilgrim
http://www.diveintopython.net/toc/index.html

A Byte of Python
Swaroop C H
http://files.swaroopch.com/python/byteofpython_120.pdf

However, for those of you who would prefer to have printed references, I recommend you take a look at Python: A Visual Quickstart Guide by Chris Fehily, or any of the series from O'Oreilly, and especially A Friendly Introduction to Numerical Analysis by Brian Bradie. Together these books cover most of what we will discuss in the course (and much more). The Bradle book should be particularly useful for those of you who want a deeper understanding of the material covered and or who anticipated making greater use of the techniques discussed for studies in your own field.

Computing Requirements: Students will be given access to the Department of Computer and Mathematical Science's computer laboratory in BV4--.

Students may also wish to install a Python interpreter on their own computer. Interpreters are freely available for a wide variety of operating systems.

Tutorials: Weekly tutorials will be held in the computer lab. Since seating is limited, students must attend the tutorial section in which they are enrolled. These tutorials are actually "practicals." And in addition to expanding on the lecture material, will involve material not covered in lectures. Students are responsible for all course material, including material discussed in the tutorials.

Assignments: There will be between three and five assignments with due dates throughout the term, Although the assignments are graded, and count towards the final grade, the primary purpose of the assignments if to provide the student with the opportunity to improve their understanding of the numerical analysis and programming techniques discussed in lectures by applying them to specific problems.

Mid-Term Test and Final Exam: There will be one two-hour mid-term test, tentatively set for mid-February, and one three-hour final exam, to be held during the exam period. Final dates will be announced later in the term.

Grading Scheme:

Assignments	30%
Midterm	25%
Final Exam	45%

HONESTY:

With the exception of exams, tests, and quizzes, you may discuss problems and homework assignments with others so long as what you turn in is your own work. Any discussions are part of the learning process; once you learn how to solve a problem you are expected to solve it by yourself, and the solution you turn in must be your own work. You must write out the solution in your own words after the discusion has ended. It is a serious academic offence to turn in any solution that has been copied from the work of someone else.