Introduction to Computational Methods in Physics

Notre Dame Physics REU / Summer 2016

Instructor: Michael Moran (`mmoran9@nd.edu`)
TA (undergraduate): Yilong Yang

Class sessions

Tuesdays & Thursdays
4:30 PM – 6:30 PM (5:00 PM – 7:00 PM for the last two sessions)
412 Jordan Hall of Science

Dates: June 7 – June 30

You may either use the lab computers or your own, so long as your personal laptop has the required software installed.

Why this course?

The physics problems you encounter in most courses can be solved with paper and pencil; physics problems in the real world typically cannot. While paper and pencil are still needed for these problems, at some stage in the process it becomes essential to make use of numerical methods of solution. Computation is also integral to interpreting the data from all but the simplest experiments. Furthermore, between the two traditional divisions of physics, theory and experiment, now lies a new approach to studying nature–simulation, or exploring the properties of physical systems (ranging from materials to galaxies) by running "experiments" on a computer.

In short, computational methods permeate physics. Computational problem solving skills are essential in graduate studies and scientific careers and in quantitative applications outside of physics as well. More immediately, computational skills can be vital to your participation in undergraduate research and future opportunities elsewhere.

Note: This course is made available to you as an opportunity for you to fill an essential gap in your physics education, or if you feel you can otherwise benefit from this course. It is possible that you *may* be able to make use of your newfound programming skills in your REU project at Notre Dame this summer, but that is not the goal of the course.

Description

In this course, you will obtain a foundation in the basic elements of scientific programming, from which you can later continue to build your scientific programming knowledge and skills. No prior knowledge of programming is required. We will be using the Python programming language, which is a powerful, modern language rapidly becoming a favorite language in several domains, including computer science instruction, scientific computing, and the commercial and industrial world.

This course is based on materials developed by Prof. Mark Caprio for Notre Dame's undergraduate Computational Methods in Physics course. You will cover much of the foundational material taught in the first half of that course. The course is a laboratory-style course, requiring advance reading of materials before the classes and full, active participation in each class.

Once we have built a solid foundation, we will move on to studying and implementing numerical techniques for scientific programming, using well-established scientific libraries. Scientific calculations frequently involve numerical calculus, numerical solutions to algebraic equations or differential equations, and numerical linear algebra operations. In addition, we will cover applications of these techniques to analyzing experimental data.

Expectations

This course, due to its condensed nature, will be more intense than a regular semester's computer science course. It will require your full engagement and commitment, including preparation outside of class. *Your participation in this course is a commitment both to yourself and to your session partners.*

If you participate in the course, it is expected that you will: participate in *all* class sessions, complete the necessary readings in advance, work constructively with your session partner, and be actively engaged in programming during class. Although it is intended that you will be able to complete all programming work during the class sessions, you should also be prepared to finish any work on your own, outside of class, in order to keep up with the class.

Textbooks

We will be using excerpts from various textbooks, both for introductory Python and numerical methods. The relevant sections are either freely available online or will be provided.

- Allen B. Downey's *Think Python: How to Think Like a Computer Scientist* (O'Reilly 2012). This book is freely available at http://thinkpython.com.
- Selected chapters from Mark Newman's *Computational Physics* (CreateSpace 2012). The relevant chapters are freely available at http://www-personal.umich.edu/~mejn/computational-physics.

Software

This course will use Python 3 for all programming. Development can be done using the IDLE integrated development environment, though this is not necessary. In addition, three scientific add-on packages (NumPy, SciPy, and matplotlib) will be used during the course, but are not necessary for the first week. Installation guidelines are posted at http://www.nd.edu/~mmoran9/reu2016, and the instructors will be available to help with any difficulties.