PY502: Computational Physics, Fall 2019

Course description
Numerical solutions, simulations, and data analysis underlie most of modern research in physics and related fields. PY502 is an introductory course that covers programming, scientific computing, high-performance computing, and algorithms commonly used in physics research. The course will begin with an introduction to programming. The goal is to learn how to develop and code algorithms from scratch using the rich syntax of `Python`. The students will also learn how to visualize data using using `Matplotlib`. Scientific computing will cover foundational algorithms in numerical calculus and linear algebra as well as modern scientific libraries that implement these algorithms (e.g. `NumPy` and `SciPy`). The main focus will be on solving optimization problems and ordinary and partial differential equations. We will also study probabilistic algorithms including Monte Carlo simulations. The course will end with a discussion of how to optimize code for best performance and how to develop code for CPU- and GPU-based parallel architectures. In addition, the students will develop deep expertise in a specific computational method of their choice by carrying out a final project.

Prerequisites
It is assumed that most students have some prior exposure to programming (not necessarily in `Python`); however, a dedicated student can succeed without prior programming experience. Solid knowledge of calculus is assumed. Basic knowledge of linear algebra, differential equations, and probability is also necessary. The course will feature many examples from different areas of physics, but specialized physics knowledge will not be necessary to develop computational solutions. It is advised that the students have access to a laptop that they can bring to class.

Teaching Staff

Instructor: Prof. Kirill Korolev	Grader: Ashish George
Email: korolev@bu.edu	Email: ashishge@bu.edu
Office: LSEB 911	Office: SCI 263
Office hours: Wednesdays 10-11	

Meeting Times and Important Dates
Class time: 11-12 in PRB 150 on Tuesdays and Thursdays and 2:30-3:50 in PRB 146 on Fridays.
Most Tuesday and Thursday classes will be lectures while most Friday classes will be programming labs.

In-class midterm: Tuesday, November 19.
Final exam: None.
Project proposals: in-class presentations on Friday, November 15.
Final project presentations: Friday, December 6, and Tuesday, December 10, during regular class time.

Students should let the instructor know if they anticipate conflicts with these dates as soon as possible.

Grading Policies
Homework: 50%
Midterm: 25%
Final project: 25%

Late assignments receive no credit.
The midterm is "closed everything", i.e. no lecture notes, books, calculators, cheat sheets, etc. are allowed.

Useful textbooks
The course is self-contained and does not follow any particular textbook. However, the students may need to consult appropriate texts depending on their prior knowledge or special interests.

- Python programming: An introduction to computer science by John Zelle; ISBN 1590282752. This is a book for those without any programming experience or knowledge of Python syntax.

- A student's guide to Python for Physical modeling by Jesse M. Kinder and Philip Nelson; ISBN 0691180571. This is another introductory book on Python. It covers elements of scientific `Python`, but does not provide in-depth discussion of programming and computer science concepts.
- Python data science handbook: Essential tools for working with data by Jake VanderPlas; 9781491912058. The book introduces iPython, Jupiter Notebooks, Matplotlib, and NumPy. It also covers many other topics that will not be discussed in class.
- Numerical methods: design, analysis, and computer implementation of algorithms by Anne Greenbaum and Timothy P. Chartier; ISBN 0691151229. This provides a very accessible introduction to numerical analysis.
- Numerical recipes: The art of scientific computing by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery; ISBN 0521880688. This is a classic book on numerical analysis.