SYLLABUS

PHY410/PHY505: Computational Physics 1

Hours: **MWF 2-2:50 PM**
Classroom: **TBD**
Instructor: Dr. Salvatore (Sal) Rappoccio
Office: 335 Fronczak
Phone: 645-6250
E-mail: srrappoc@buffalo.edu
Office Hours: Wed 3-5, and by appointment

This course integrates numerical analysis and computer programming in C++, with python as an optional second language, to study a variety of problems in classical, quantum, and statistical physics. The course will cover numerical algorithms for root finding, interpolation, matrix inversion, numerical differentiation, and quadrature, data analysis, Fourier transformation, and computer graphics. If time permits, we will also have an introduction to quantum computing. Numerical analysis topics will include solution of linear and nonlinear differential equations, boundary-value and eigenvalue problems, and Monte Carlo simulation. The computational content of the course will be organized in the following topics: (1) Data Analysis, (2) Basic Numerical Algorithms, (3) Linear Algebra, (4) Solving Nonlinear Equations, (5) Ordinary Differential Equations, (6) Partial Differential Equations, (7) Probablistic Methods, and (8) Quantum Computing. Depending on the preparation of the class, the topic may be pushed to the second semester to make more room for introductory programming practice.

The software for the course will be hosted on github.

PREREQUISITES AND BASIC RESOURCES:
This course assumes familiarity with undergraduate physics at the junior/senior level. You should have passed PHY 301, PHY 401, and PHY 403, or equivalent courses, or be taking them concurrently. If you are not a physics major, a strong background in undergraduate mathematics or computer science should suffice if you spend extra time to learn the physics background required for each topic, although you should be familiar with ordinary and partial differential equations at the very least.

Familiarity with a modern programming language is required (C++/Java/Fortran/`python`/etc). Programming mainly with C++ and `python` will be covered in the first 4-8 weeks of lecture. If you are not familiar with C++ or `python` you should spend extra time very early in the course to bring yourself up to speed. Depending on experiences of the class, we will spend more or less time on introductions to programming. You will also need to know how to compile and execute your code on your chosen platform. For instance, familiarity with `bash`, `tcsh`, or `zsh` for Linux/Unix/Macintosh, or `cygwin` for Windows. It is useful to have sufficient familiarity with `python` to use this for shell scripts also. You're also welcome to combine C++ and `python` with some existing tools such as SWIG.

The following are helpful introductory resources:
- http://www.physics.buffalo.edu/phy410-505/ Previous years' course site
- Programming - Principles and Practice Using C++ by Stroustrup
- http://www.python.org Python programming language official website

- http://docs.python.org/2/tutorial/ Python tutorial
- http://www.swig.org : SWIG for combining C++ and python

REQUIRED MATERIALS:
Personal laptops are recommended for this class. The primary method of distribution of software is via Docker container, so please ensure you can use docker if you use a personal laptop. Alternatively you can use Windows with cygwin, Macintosh (OS X 10.5 or later), or Linux (e.g. Ubuntu). All required software for this course can be downloaded for free. Detailed lecture notes and online references will be provided. The Windows 10 operating systems may use the bash shell functionality, although that is not officially supported at this time.

The required textbooks for programming are free of charge :

- Fundamentals of C++ Programming by Richard Halterman
    - Example code at https://github.com/halterman/CppBook-SourceCode
- https://docs.python.org/3.1/tutorial/ : The Python Tutorial

You are also required to have access to Numerical Recipes. The latest version does cost money but is a worthwhile investment for your career, while older versions of NR are free.

- Numerical Recipes in C++
    - Earlier online version of NR for free

You may also find this resource useful :

- Numerical Methods for Physics by Alexander Garcia

The course website is at UBLearns :

- http://ublearns.buffalo.edu/ UBLearns course site

We will also use slack for this course. You are welcome to participate remotely or in person.

COMPILERS, SHELLS AND TOOLS :

Windows :

- http://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/ BASH shell for Windows10
- http://mingw-w64.sourceforge.net GCC for Windows (included in `cygwin`)

Macintosh :

- https://developer.apple.com/xcode/ : XCode

Linux :

- http://gcc.gnu.org : GCC (included in most distributions anyway)

Shells :

- http://www.gnu.org/software/bash/manual/bashref.html : `bash` shell
- http://www.tcsh.org/Welcome : `tcsh` shell
- http://www.zsh.org : `zsh` shell
- http://www.gnu.org/software/make/ : gnu make

LaTeX :

- http://www.tug.org/texlive/ : TeX live

Plotting tools :

- http://matplotlib.org : matplotlib
- http://www.gnuplot.info : gnuplot
- http://root.cern.ch/ : ROOT

Editors :

- http://www.gnu.org/software/emacs/ : emacs
- http://www.vim.org : VIM
- https://developer.apple.com/xcode/ : XCode

Version Control Software :

- http://github.com : git

SCHEDULE:
The course is scheduled MWF 2-2:50 PM. Homework will be regularly assigned (~weekly). There is a take-home midterm and final exam.

EXPECTATION
To succeed in this course you should read the lecture notes and posted materials, attend class and participate actively in discussion and quizzes, complete the homework assignments on time, and take the midterm and final exams. Exceptions will be made for documented medical reasons or major emergencies.

If you are having difficulty with the course material, it is best to be proactive and contact me directly, either in office hours or by appointment. Discussing difficulty beforehand is encouraged, but asking for special consideration after the fact is not usually helpful.

GRADING:
Grades will be based on your scores on homework (50%), one in-class midterm (25%), and a take-home final exam (25%). Graduate students and undergraduates will be graded separately.
The lowest homework score will be dropped from consideration to accommodate personal situations such as illnesses or missed classes.

MIDTERM: Mid semester (take home
FINAL: Last week of classes (take home).

ACADEMIC INTEGRITY
Academic integrity is a core value underlying all scholarly activity in the Department of Physics. Please review UB undergraduate policy at http://undergrad-catalog.buffalo.edu/policies/course/

integrity.shtml or graduate policy in http://www.grad.buffalo.edu/policies/academic_integrity.pdf. You are encouraged to discuss class material and assignments with your colleagues (with acknowledgment of who you worked with on your assignment). However, you should code and run your simulations yourself, and your homework writeup must be entirely your own effort. If you copy and/or modify code from any source for your assignments you should acknowledge this with an appropriate citation in your writeup.

STUDENTS WITH DISABILITIES

If you have a disability, (physical or psychological) and require reasonable accommodations to enable you to participate in this course, such as note takers, readers, or extended time on exams and assignments, please contact the Office of Disability Services, 25 Capen Hall, 645-2608, http://www.student-affairs.buffalo.edu/ods/, and also see me me during the first two weeks of class. ODS will provide you with information and review appropriate arrangements for reasonable accommodations.

Undergraduate Learning Outcomes

TOPIC UNITS	**LEARNING OUTCOMES** Students are expected to master the following:	**OUTCOME ASSESSMENT** Learning on topics is assessed as follows:
Programming and Technical Computing	Introduction to UNIX environment, compilation, programming in C++ and python, debugging.[3]	Homework, midterm, exam
Data analysis	plotting, data fitting, analyzing large datasets, shell scripts and compilation [1,2,3]	Homework, midterm, final
Basic numerical algorithms	Derivatives, quadrature, interpolation, root-finding, special functions, the FFT algorithm [2,5]	Homework, midterm
Linear algebra	Matrices, BLAS algorithms, solving linear algebraic equations, programming with objects [2,5]	Homework, midterm

Solving nonlinear equations	Minimization and maximization of functions, multi-dimensional root finding, nonlinear models of data [2,5]	Homework, final
Ordinary differential equations	Initial value and boundary value problems, the Kepler and 3-body problems, chaotic dynamics in nonlinear systems, quantum eigenfunctions and eigenvalues [2,5]	Homework, final
Partial differential equations	Elliptic, parabolic and hyperbolic equations, Poisson's equation in electrostatics, wave motion, spectral methods, quantum wavepacket motion. [2,5]	Homework, final
Probabilistic methods	Random numbers, random walks, polymer dynamics, the Metropolis algorithm, Monte Carlo simulation of the hard disk gas and the Ising model [2,5]	Homework, final
Quantum computing	Introduction [2,5]	Homework, final

The bracketed numbers in the 2nd column give the correspondence to the Physics Department's undergraduate curriculum goals: [1] The basic laws of physics; [2] Critical thinking and problem solving; [3] Laboratory skills; [4] General knowledge of the development of physics; [5] Contemporary areas of physics inquiry; [6] Written and oral communication skills. Note that not all courses emphasize all of the above goals.

Graduate Learning Outcomes

TOPIC UNITS	**LEARNING OUTCOMES** Students are expected to master the following:	**OUTCOME ASSESSMENT** Learning on topics is assessed as follows:
Programming and Technical Computing	Introduction to UNIX environment, compilation, programming in C++ and python, debugging.[4]	Homework, midterm, exam

Data analysis	C++ and python coding, plotting, data fitting, analyzing large datasets, shell scripts and compilation [1,2,4]	Homework, midterm, final
Basic numerical algorithms	Derivatives, quadrature, interpolation, root-finding, special functions, the FFT algorithm [2,4]	Homework, midterm
Linear algebra	Matrices, BLAS algorithms, solving linear algebraic equations, programming with objects [2,4]	Homework, midterm
Solving nonlinear equations	Minimization and maximization of functions, multi-dimensional root finding, nonlinear models of data [2,4]	Homework, final
Ordinary differential equations	Initial value and boundary value problems, the Kepler and 3-body problems, chaotic dynamics in nonlinear systems, quantum eigenfunctions and eigenvalues [2,4]	Homework, final
Partial differential equations	Elliptic, parabolic and hyperbolic equations, Poisson's equation in electrostatics, wave motion, spectral methods, quantum wavepacket motion. [2,4]	Homework, final
Probabilistic methods	Random numbers, random walks, polymer dynamics, the Metropolis algorithm, Monte Carlo simulation of the hard disk gas and the Ising model [2,4]	Homework, final
Quantum computing	Introduction [2,5]	Homework, final

The bracketed numbers in the 2nd column give the correspondence to the Physics Department's graduate curriculum goals: [1] The basic laws of physics; [2] Advanced knowledge in a specialty area; [3] Broad knowledge of physics topics outside the specialty area; [4] In-depth scientific research skills; [5] Teaching and communication skills. Note that not all courses emphasize all of the above goals.