Phys3274 Computational Physics

Instructor:


```
E.S. Swanson
404 Allen Hall
4-9057       swansone@pitt.edu     http://fafnir.phyast.pitt.edu/CompPhys/
```


class meets Tuesday and Thursday, 2:30-3:45, 106 Allen Hall.

Office Hours: Tuesday and Thursday, 3:45 - 5:00. You can stop by anytime, but it might be safer to make an appointment.

Course Description:

Physics 3274 is a graduate course on computational physics. It aims to develop or reinforce programming skills, numerical analysis skills, familiarity with some important problems in computational physics, and their methods of solution. The course will employ the C++ language, hence some familiarity with C (or better, C++) is recommended. Primary topics to be discussed are (1) C++, including the concepts of object-oriented programming; (2) numerical techniques, including essential methods of integration, discretization, Monte Carlo, and diagonalization; (3) physics, including percolation, chaos, classical dynamics, many-body systems, spin systems, continuum mechanics, quantum mechanics, and data modelling. Additional topics such as parallel computing and sockets will also be covered.

Course Objectives:

By the end of PY3274 the student will be able to:

- assess the feasibilty of computational solutions to complex physics problems
- design an efficient approach to solving complex physics problems
- employ modern design standards in solving complex physics problems
- locate resources that permit the solution of complex physics problems
- produce publication quality graphs
- produce publication quality written reports

Text:

- J.F. Boudreau and E.S. Swanson, *Applied Computational Physics* (Oxford University Press, 2017).

A standard reference in the field that I recommend buying is

- W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vettering *Numerical Recipes*

This book used to be acquired for its code, but these days it will be more valuable for the discussion of techniques and algorithms. The third edition uses C++.

Supplementary Texts:

- N.J. Giordano and H. Nakanishi, *Computational Physics* [undergraduate level]
- C.M. Bender and S.A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers* [a classic

for methods]
- W.R. Gibbs, *Computation in Modern Physics* [graduate level, but very short]
- J.P. Sethna, *Entropy, Order Parameters, and Complexity* [wonderful introduction to the modern view of complex classical systems. Computational problems throughout. Available online!]

Marking Scheme:

0.7 assignments + 0.3 final project

Syllabus:

- Introduction
- Brief Introduction to C++
  - compiling and linking, libraries
  - communicating with programs
  - style guide
  - round-off error
  - encapsulation
  - classes
  - header files
  - function templates
  - class templates
- Interpolation, Extrapolation, and Quadrature
  - Lagrange interpolating polynomial
  - splines
  - Shanks and Richardson extrapolation
  - Taylor series, Pade approximants, continued fractions
  - simple quadrature, Simpsonâ€™s rule, Gaussian quadrature
- Monte Carlo Methods
  - random variates
  - Monte Carlo integration
  - Markov chain Monte Carlo
  - the heat bath algorithm, Gibbs sampling
- Percolation and Universality
  - percolation
  - cluster algorithm
  - scaling laws and critical exponents
  - universality and the renormalization group
- Parallel Computing
  - parallel computing paradigms
  - MPI, openMP
  - C++ concurrency library
  - [forks and sockets]
- Ordinary differential equations
  - Euler method
  - Runge-Kutta method
  - adaptive step size
  - symplectic integration
- Chaos
  - nonlinear dynamics

    - iterative maps
    - the Lyapunov exponent
    - KAM theorem
- Introduction to data modelling
    - function minimization
    - unbinned maximum likelihood
- Molecular Dynamics
    - statistical mechanics
    - the Verlet method and simple gasses
    - multiscale systems, constrained dynamics
    - gravitational systems
    - the Barnes-Hut algorithm
    - particle-mesh methods
- Continuum Dynamics
    - partial differential equation
    - initial value problems
    - time-dependent Schroedinger equation
    - boundary value problems
    - fast Fourier transform
    - finite element methods
- Classical Spin Systems
    - finite temperature systems
    - the Ising, Potts, XY models
    - first, second, infinite order phase transitions
- Quantum Mechanics I
    - simple bound states, discretization, momentum space,
    - quantum Monte Carlo methods
    - scattering and the T-matrix

depending on time and interest, we may cover some of the following:

- Quantum Mechanics II
    - atoms
    - molecules
    - Hartree-Fock theory
    - density functional theory
- Quantum Spin Systems
    - magnetism
    - the Lanczos algorithm
- Quantum Field Theory
    - the path integral
    - phi-4 theory
    - Abelian gauge theory
    - nonAbelian gauge theory
    - fermions
- Artificial Intelligence
    - neural networks
    - learning algorithms
    - categorization and pattern recognition
    - convolution networks

Assignments:

- to be posted...

Prerequisites

This course covers *a lot* of material, so, while it is self-contained, it is best to come prepared. This means

- remembering your undergraduate classical and quantum physics
- basic knowledge of the unix operating system
- knowing C at a decent level. (these things, for example, should be familiar to you: call by value vs. call by reference, scope, arrays, program flow)
- knowing how to latex
- knowing how to create decent quality graphs

Don't panic: help will be available for the latter two; we will discuss C++ in class (but you should do more reading out of class); and I will provide physics reminders as we go.

Things to do before class begins:

- install Linux or OS X
- install C++ compiler (Xcode, or gcc)
- install the Qat library (gsl, Eigen3, coin3D + SoQt,...), available here: qat.pitt.edu .
- install Latex (brew cask install mactex)
- install a publication quality graph creator [brew install gnuplot --with-x11 OR brew install gnuplot --with-qt]
- install Qt (optional, but nice) [brew cask install qt-creator]
- install ACP example code (to appear ~ September) (optional, but helpful)
- install Minuit2 (needed by October) [go here. Then configure; make; make install]
- install eclipse (not required, but nice)

Resources:

Qt

- www.qt.io/developers [learn some Qt!]
- wiki.qt.io/Qt_for_beginners

C++

- www.learncpp.com [good place to start]
- cplusplus.com [excellent reference in tutorials]
- Bronson, G. J. (2013). C++ for scientists and engineers. Cengage Learning.
- Capper, D.M. (1994). The C++ programming language for scientists, engineers, and mathematicians. Springer-Verlag.
- Meyers, S. (2005). Effective C++. Available online

miscellaneous

- eigen.tuxfamily.org [the excellent Eigen3 library for linear algebra]
- www.gnu.org/software/gsl/html/index.html [gnu scientific library]
- www.boost.org [boost additions to C++]

latex

- Latex tutorial [html]
- Latex tutorial [html]
- Latex tutorial [pdf]
- revtex4 home page [html]
- sample paper [tex]
- sample paper [pdf]

Academic Integrity:

Students in this course will be expected to comply with the University of Pittsburgh's Policy on Academic Integrity. Any student suspected of violating this obligation for any reason during the semester will be required to participate in the procedural process, initiated at the instructor level, as outlined in the University Guidelines on Academic Integrity. This may include, but is not limited to, the confiscation of the examination of any individual suspected of violating University Policy. Furthermore, no student may bring any unauthorized materials to an exam, including dictionaries and programmable calculators.

Disability Services:

If you have a disability for which you are or may be requesting an accommodation, you are encouraged to contact both your instructor and Disability Resources and Services (DRS), 140 William Pitt Union, (412) 648-7890, drsrecep@pitt.edu, (412) 228-5347 for P3 ASL users, as early as possible in the term. DRS will verify your disability and determine reasonable accommodations for this course.