Instructor:	Olga Goulko	Lectures:	Tu&Th 2-3:15pm, M3-732
Office hours:	Tu&Th 3:30-5pm	Lab:	Fri 2-3:50pm, M3-732
Office:	ISC-1-1180	Phone: 617-287-6932	
Email:	`olga.goulko@umb.edu`	Homework due:	Fri 2pm

This Python-based course provides an introduction to the uses of computational tools in physics and other natural sciences. It covers the basics of scientific programming, including a review of Python and general programming practices, numerical analysis (linear and nonlinear equations, data fitting, numerical integration, differential equations, and Fourier methods), and simulations of physical systems, including Monte Carlo and molecular dynamics simulations. The analysis and visualization of numerical and experimental data will play an important role. The course is a combination of lectures on techniques hands-on programming exercises in the labs. This course is combined with PHYSIC 362. Credits: 4

Students are expected to have a background in undergraduate level science and mathematics, as well as a basic knowledge of programming. Students who are not matriculated in the Applied Physics graduate program may enroll with the permission of the instructor. The course material is intended to be accessible to a scientifically diverse student body, including students in mathematics, computer science, physical and other natural sciences.

Expected student learning outcomes

- being able to turn mathematical equations into code
- being able to make use of standard algorithms for numerical modeling of physical systems
- being able to create simulations of physical systems
- being able to use computational methods to read, write and manipulate data
- becoming familiar with the Python programming language and useful programming tools
- acquiring good programming style for scientific problems
- being able to create useful and informative scientific plots
- recognizing potential uses of computational methods for research problems

These skills are essential for all areas of physics and will be useful for your graduate research and courses, as well as academic and professional careers.

Course topics

The topics and timing may change as the semester develops. The exams are tentatively scheduled during lectures 14 and 29.

1. Scientific computing basics (Week 1-3)
   - Role of computational physics (Lecture 1)
   - Good programming practices (Lecture 2)
   - Review of Python basics (Lecture 3-4)
   - Algorithms, computational efficiency, numerical errors and accuracy (Lecture 5)
2. Numerical methods (Week 3-11)
   - Linear algebra: matrix operations, linear equations, eigenvalue problems (Lecture 6-8)
   - Non-linear equations, root finding, function optimization (Lecture 9-10)
   - Data fitting: interpolation and extrapolation, splines, least-squares (Lecture 11-16)

- Numerical integration (Lecture 17-18)
- Ordinary differential equations: Euler, midpoint, Runge-Kutta (Lecture 19-20)
- (Fast) Fourier transform (Lecture 21-22)

3. Simulations of physical systems (Week 12-15)

   - Random Systems: random number generation, random walks, diffusion, cluster growth (Lecture 23)
   - Monte Carlo simulations: integration, importance sampling, Markov chains, Metropolis algorithm (Lecture 24-25)
   - Simulated annealing: function optimization, spin systems, parallel tempering (Lecture 26)
   - Molecular dynamics simulations: Verlet algorithm, atomic collisions (Lecture 27-29)

Book suggestions

There is no required textbook. The course will be self-contained, but you are encouraged to read more about the material in textbooks and to make use of the many available (and free!) online resources, tutorials etc. Specific references and recommendations will be given throughout the course. Here is a list of books that me and others have found useful:

- "Computational Physics" by Mark Newman (guide to Python in computational physics)
- "Numerical Recipes" by W. H. Press *et al.* (covers very comprehensibly a vast range of numerical topics; 3rd edition is in C++; older editions, e.g. in C, are freely available online)
- "Clean Code" by Robert C. Martin (great book on good programming practices)

Lectures and Labs

The course lectures will be a combination of lectures on techniques and example programs. The labs will be less formal than the lectures with main focus on hands-on programming. We will also further discuss and review the course material and homework, especially the more difficult topics. You can consult with the instructor and other students on the concepts and problems, or work together on programming. The labs will help you solve the homework assignments.

Computations

Python will be our default programming language. Laboratory and homework will be done on your own laptops or on the computers in Web lab M3-732. You will have access to a remote working environment with pre-installed software. Early work in the labs will help you get familiar with the needed software and tools, and ensure that everyone has a working programming environment.

Homework

There will be weekly homework assignments, which need to be submitted by the deadline via Gradescope. Graduate homework assignments will contain additional problems. Instructions on the homework submission will be given on the first assignment. In addition, there will be short weekly pre-class warm-up exercises, to be submitted on Blackboard one hour before the first lecture of the week. These will be graded based on (honest) participation only and will help you get prepared for the week's lectures.

Project

The project is an opportunity to apply the methods learned in the course to a physical system that interests you (graduate project topics will be more complex). It will require writing a computer program and presenting the problem, methods, results, and conclusions during the finals slot at the end of the course.

Grading

The course grade will be calculated as follows:

- warm-up exercises: 5%
- homework: 35%
- two exams: 40% (20% each)
- project: 20%

Course grade percentages (may be adjusted):

A:	≥92%	A-:	92%-88%	B+:	88%-84%	B:	84%-80%
B-:	80%-76%	C+:	76%-72%	C:	72%-68%	F:	<68%

Accommodations

UMass Boston is committed to providing reasonable academic accommodations for all students with disabilities. Please contact the Ross Center for Disability Services (`ross.center@umb.edu`, 617-287-7430) for recommendations for specific accommodations if needed. If you already have a recommendation, please contact me as soon as possible, preferably within the first two weeks of classes, so that we can work out the best way to support you.

Student mental health

If you or someone you know experiences academic stress, difficult life events, or feelings of anxiety or depression, we encourage you to seek support. Helpful, effective resources are available via the University Health Services. The UHS Counseling Center can be reached at 617-287-5690 (including after-hours and on weekends). Whether or not you are a current patient at the center, you will be able to access telehealth crisis support. Please know that the Counseling Center and university are working to provide remote support for students during this difficult time. More information is available online at
`https://www.umb.edu/healthservices/counseling_center`

Student conduct

Students are required to adhere to the University Policy on Academic Standards and Cheating, to the University Statement on Plagiarism and the Documentation of Written Work, and to the Student Code of Conduct. The Student Code of Conduct and Instructional Setting Conduct Policy are available online.