

Syllabus for PY 421 “Introduction to Computational Physics”

Physics Department – Spring Term 2024

Instructor: Claudio Rebbi

Lectures: Tuesday and Thursday, 11:00AM to 12:15PM, room KCB 104

Discussion sessions: Friday, 3:35PM to 4:25PM, room CAS 310

This course introduces students to computational techniques for the solution of research problems in physics. After preliminary lectures on the basics of computing, the course proceeds toward modern methods of computational physics. In particular the data parallel paradigm and graphics rendering are illustrated through computationally intensive applications. All of the techniques are presented in the context of specific projects. For some problems, students are asked to play the role of application scientists, working with the computer in an interactive manner to perform a variety of numerical experiments. For other problems, students are required to take an active part in designing the code that will be used for the solution. Emphasis in all cases will be on the importance that the proper formulation of a problem plays for its ultimate computational solution.

Applications Problems:

Image manipulations.— Some simple operations are implemented by parallel algorithms on a two-dimensional array representing the intensity values of the pixels in a monochrome image. A relationship is established between the operations performed by computer (e.g. blurring of the image) on the discrete set of image data and fundamentals equations of mathematical physics for continuum fields (e.g. the diffusion equation).

Calculation of the Electrostatic Potential.— Various methods for solving Poisson’s equation are evaluated in terms of their suitability for the data parallel paradigm. Students are asked to write the code for some basic iterative methods of solution on a parallel machine. Numerical experiments are used to demonstrate the occurrence of critical slowing down. Multigrid methods and other algorithms for overcoming critical slowing down are analyzed for implementation in a data parallel environment. Advanced visualization software is presented.

Solution of the One-Dimensional Schrödinger Equation.— The time-dependent Schrödinger equation in one space dimension is solved by means of an algorithm that uses repeated fast Fourier transforms in order to preserve the norm of the wave function. The complex wave function is represented graphically by means of a color map whereby its phase is encoded by the color circle. Here the student is provided with code and asked to use it as an application scientist, experimenting with a wide range of initial data.

Simulation of Phase Transitions in Spin Systems.— Stochastic algorithms (Monte Carlo methods) are introduced for the simulation of models, such as the Ising model and the N-state Potts model, which exhibit interesting thermodynamical behavior and phase transitions. Programs for the implementation of such algorithms on a parallel machine are analyzed. Students then use these programs to perform large scale numerical experiments on phase transitions. Particular attention is given to the management of the large base of data that the simulations generate and to developing the auxiliary programs required for the analysis and interpretation of the data.

Molecular Dynamics Simulations.— Students are asked to write serial code for the implementation of molecular dynamics simulations. Following optimization and numerical experimentation with this code, the students and instructor collaborate to write highly efficient code for the simulation on a parallel machine. Performance for the serial and parallel modes of computation are compared and evaluated.

Requirements

PY 351 or equivalent, or consent of the instructor.

Schedule

Classes will begin on Thursday January 18 and end on Tuesday April 30. Please see the appendix for the schedule of exams and other relevant dates.

The schedule of lectures shown below is tentative. The actual pace of lectures will be tailored to the students' needs.

Week 1: Basics of computer usage and of programming in C. Numerical differentiation and integration.

Week 2: Programming in C, continued. Writing to a file. Numerical solution of simple ordinary differential equations, e.g. for projectile motion with air resistance. Graphing of the results.

Week 3: Programming in C and elementary graphics rendering. Solution of the one-dimensional wave equation with the leapfrog algorithm and visualization.

Week 4: Elements of Fortran. Array constructs and operations with arrays.

Week 5: Blurring of an image. Discretized Laplacian. The diffusion equation. Review of the finite and continuum Fourier transform and the fast Fourier transform (FFT).

Week 6: Eigenvectors of the discretized Laplacian. Instability of the algorithm for solving the diffusion equation. Unblurring the image.

Week 7: Solution of the equation for the electrostatic potential in presence of conductors. Relaxation algorithms and their relation with a diffusion process. Critical slowing down. Mention of multiscale methods.

Week 8: Solution of the one-dimensional time-dependent Schrödinger equation by a split operator formalism and the FFT. Visualization of the evolution. Importance of preserving unitarity.

Week 9: Generation of (pseudo)random numbers. Monte Carlo integration.

Week 10: Monte Carlo simulation of the Ising model. Parallel computing and MPI (message passing interface).

Week 11: The path integral formulation of quantum mechanics. (*Note:* this uses the same mathematical ingredients already introduced for the numerical solution of the Schrödinger equation, which makes the whole notion of path integral quite natural and easy to understand.)

Week 12: Stochastic simulation of the path integral.

Week 13: Molecular dynamics simulation of a system of atoms interacting with Lennard-Jones potentials.

Week 14 and last class: Make-up and summary.

Readings

Mandatory reading:

Class notes and computer programs, distributed in class.

Optional readings:

1. “Numerical Recipies” by W. Press, B. Flannery, S. Teukolsky and W. Vetterling (Cambridge University Press)
2. “The C Programming Language” by Brian W. Kernighan and Dennis M. Ritchie (Prentice Hall)
3. “Fortran 90/2003 Explained” by Michael Metcalf, John K. Reid and Malcolm Cohen (Oxford U. Press)

Grading

Credit for the course will be based on the homework assignments, two midterm exams, a final exam, and a term project. See the notes below for details on how the scores on these course components are converted into a final grade. Homework assignments will typically be given weekly, but occasionally the students will be allowed extra time to complete a more elaborated assignment before a new one is issued. The homework assignments will have to be carried out on the computer and will range from writing simple programs, to the application of already developed programs to the study of complex physical systems and the implementation of specific parallel algorithms, to the design of a parallel code to solve some challenging physics problem. The exams will typically consist of two (for the midterm exams) or three (for the final exam) word problems and one problem which will have to be done on the computer.

Please see below the very important note about homework assignments.

Notes:

- (a) Since there is no established textbook to cover all the topics in the syllabus for PY 421, class notes have been prepared and will be distributed. Software (i.e. code) will also be distributed, which should be considered as an integral part of the instruction material.

(b) Homework assignments are meant to be a continuation of the teaching material and not tests to see how the students are doing. The midterm and final exams are reserved for that. Students can count on receiving at least 90/100 for all the assignments they turned in with a serious attempt to solve the problems, even if the solutions are incorrect. The way this will work is as follows. The assignment problems will be initially graded in the same way as the problems in the exams, in order to give the students feedback on how they have been doing. However at the end of the course the assignment scores will be regraded in such a way that the regraded scores will be no less than 90/100, provided that a student returned an attempted solution to all the problems. For example, if the assignment consisted of three problems for which the student received 25, 17, and 20 points, which in an exam would give a score of 62/100, a base of 30 points (90 divided over the three problems) will be assigned to each problem and one tenth of the original points will be added to it, making the points on the three problems 32.5, 31.7, 32, with a total score of 96.2. **However, no points will be awarded to a problem for which a student did not submit a fair attempt to solve it or did not submit a solution.** Please note that the scale for the conversion of the overall course score to a letter grade is based on the assumption that the students will have homework assignment scores not less than 90/100. Returning incomplete assignments, or, even worse, failing to return assignments can lower the overall assignment score and be very detrimental for the final grade.

(c) Grades: The scores on homework assignments, midterm exams, final exam, and term project, normalized to 100, will be added with weights 0.3 for the homework, 0.1 for each of the midterms, 0.3 for the final, and 0.3 for the term project, not to exceed 100 (the sum of the weights purposely equals 1.1). The total will determine the final grade as follows:

$$\begin{aligned} 94 \leq f_s \leq 100 : & A; & 88 \leq f_s < 94 : & A-; \\ 82 \leq f_s < 88 : & B+; & 76 \leq f_s < 82 : & B; & 70 \leq f_s < 76 : & B-; \\ 64 \leq f_s < 70 : & C+; & 58 \leq f_s < 64 : & C; & 52 \leq f_s < 58 : & C-; \\ 0 \leq f_s < 52 : & F; \end{aligned}$$

Addendum

This addendum addresses a list of standard questions that students may have about the course.

1. Instructor's office address and phone number:

Claudio Rebbi - office: PRB 551, 3 Cummington St.; phone number: (617)353-9058; email: rebbi@bu.edu

2. Office hours: TBA and by appointment.
3. Textbook and other course material: as mentioned in the main body of the syllabus, no textbooks are required. Material for the course consists in the lecture notes and the programs which the students will find in the `~rebbi/courseware/notes` and `~rebbi/courseware/code` directories on the CAS cluster.

4. Required readings: students are expected to read and study all lecture notes as well as the programs distributed to the class.
5. Web site: this course has no web site. All the students have accounts on the CAS cluster and can retrieve course material there as explained in 3. The CAS lab has a web site <https://www.bu.edu/casit/computer-labs/> which students can consult for schedule and other information.
6. Classes and other events which students are expected to attend: students should attend all lectures and discussion sessions.
7. Policy regarding absences: students who cannot attend a lecture should send an email to the instructor, before the lecture if at all possible, to justify their absence. The accumulation of more than 3 unjustified absences may lead to a one-step reduction in the final grade (A → A-, A- → B+ etc.).
8. Dates and times of examinations:
the **first midterm exam** will be held in class on **Thursday, March 7**;
the **second midterm exam** will be held in class on **Thursday, April 4**.
The **final exam** will be held during the final exams period, May 6 to May 10, at a date to be announced.
The deadline for submitting the **term project** will be the last day of classes, **Wednesday May 1 by the end of the day**. It is a strict deadline.
9. Grading criteria: already specified in detail in the main body of the syllabus.
10. Late work: homework assignments must be returned by the deadline stated in the assignment. Students may request an extension for valid reasons by sending an email message to the instructor before the deadline.
11. Independence of work: students should do their homework assignments by themselves. They may use help (from fellow students, faculty, IT consultants etc.), however, on technical matters such as programming languages, usage of computer tools etc.
12. Students' responsibility: Students should know and understand the provisions of the CAS Academic Conduct Code. (Copies are available in room CAS 105). Cases of suspected academic misconduct will be referred to the Dean's Office.
13. Schedule of lectures: the main body of the syllabus provides a general outline of the course. However, the material that will be treated in class may depart to some extent from that outline to accommodate the pace and interests of the students. Homework will be assigned at one to two week intervals, with the exact dates dependent on the pace of the course. Students will be given at least one week to complete their assignments.
14. Relevant dates:
The first day of classes is Thursday January 18.

Monday February 19 and Monday April 15 are holidays. The University follows a Monday schedule of classes on Wednesday February 21.

The Spring Recess is from Saturday March 9 to Sunday March 17.

The last day of classes is Wednesday May 1.

Final exams will begin on Monday May 6 and end on Friday May 10.