Computational Laboratory in Statistical Mechanics

PHYS ????? / Spring 2020 / 2 credits

Sushrut Ghonge
Office TBD
sghonge@nd.edu

Kevin Howard
103 Nieuwland Science Hall
khoward5@nd.edu

Prof. Mark Caprio
209 Nieuwland Science Hall
mcaprio@nd.edu

Laboratory meetings

Location/Time:TBD

Description

Why this course?

Course structure

This is a project-based course, built around independent readings, group meetings, and semi-independent work in loose collaboration with lab partners. The meetings will be very much in the spirit of research group meetings. They will include: seminar-style discussion of readings in preparation for each project, brainstorming and outlining approaches for accomplishing the goals of the project, status updates and exchange of ideas as challenges arise, and sharing of results and insights after each project is completed. A portion of each session will be open time dedicated to working on the current project, in a collaborative setting. Scheduling will be flexible, as we determine the most efficient scheme for working on each stage of the project; in initial stages of a project, we will often spend more meeting time discussing and planning, while in later stages we will devote more time to development.

What you have learned so far

From Computational Methods in Physics, you acquired a basic foundation in the three fundamental elements of computational problem solving:

(1) *Programming*. You are now able to speak the language of the computer. You are able to plan, code, and debug a programming project systematically and with confidence. You are now accustomed to writing programs which don't merely "work" but which are well-structured, divided into logical subunits, and clearly documented. Your programs are therefore straightforward to debug, and you write your code in a general way so that you can easily reuse and extend it as your scientific goals expand.

(2) *Numerical methods*. You have also developed proficiency with many of the numerical methods used for doing the calculations in scientific problems. Your initial introduction focused on rootfinding, numerical integration, integration of differential equations, evaluation of special functions, and an introduction to numerical linear algebra. However, more important than the *specific* numerical methods you learned in Computational Methods in Physics is the fact that you now have a general sense of the *use and limitations* of numerical methods. You have demonstrated that you can read about a new numerical algorithm on your own, implement code for this algorithm, and investigate the numerical errors (algorithmic and roundoff) and stability of the algorithm.

(3) *Translation*. The least tangible, but arguably most important, aspect of solving a problem computationally is the step of translation. That is, how do we combine the various ingredients we learned above, to solve an actual problem? We must take a general physical framework (such as Newtonian mechanics or the Schrödinger equation) and translate this into a concrete set of algorithms and calculations, to be carried out numerically, for the given problem. Which equation or equations is most suitable for solution with the computer, and how can we best set about doing this? Computational problem solving, as with any problem solving, must largely be learned from examples and experience. While you gained some initial exposure in Computational Methods in Physics, translation will be the focus of the Computational Laboratory in Statistical Mechanics.

What we will learn in this laboratory

> Computation is an integral part of modern science, and the ability to exploit effectively the power offered by computers is therefore essential to a working physicist. The proper application of a computer to modeling physical systems is far more than blind "number crunching", and the successful computational physicist draws on a balanced mix of analytically soluble examples, physical intuition, and numerical work to solve problems that are otherwise intractable. — *Steven E. Koonin*

Prerequisites/corequisites

Organization

Resources

Software

This course will be based on Python 3, the language you used in Computational Methods in Physics. We will also make use of some of the well-developed scientific libraries for Python: the numerical linear algebra library NumPy, the scientific library SciPy, and the plotting library matplotlib. You will need to have these installed on your laptop.

Note: In practice, if you wish to write your code in Python 2.7, this will be permissible. There are certain features in Python 3 which we will often take for granted in this class (division of integers being a prime example) which you will be responsible for accounting for in your submitted code. Do so at your own risk! Further, one should always include the language version utilized for a project within the document header.

Projects

This course will be structured around projects, each of several weeks' duration. The life-cycle of a project will be as follows:

> (1) Readings on the physical problem and any relevant numerical methods **will be required**, with reading assessments as appropriate. Since the basic physical principles will typically have already been covered in Quantum Mechanics I or II, the readings will focus on the specific problem at hand and the computational details.
>
> (2) A guided discussion of the reading will provide opportunities for clarification and questions. This will be a seminar-style discussion — you will have opportunities to present and explain portions of the reading to your peers and/or raise questions on the parts which are still not clear to you. Discussion will focus on putting this physical problem in context and on planning an approach for carrying out the project. Getting to the "correct answer" is not the end goal, but just the start! What *insight* can we gain about the physical system? How does its behavior depend upon the parameters of the problem?
>
> An important step here will be reviewing or working out *analytic* solutions to test cases which we will use to benchmark our codes. We will also usually need to transform the mathematical formulation problem to a form which is more amenable to numerical solution (*e.g.*, transformation to convenient units, change of variable).
>
> (3) You will continue to work on carrying out the project. Planning and coding are the "obvious" parts. But this is research we are doing. Just as important is *testing and validation* — your code will not give the correct answer the first time! How do you know when you can finally believe your results? (Can we reproduce the analytical results we derived ahead of time?)
>
> Based on the success of the partner system in Computational Methods in Physics, you will continue to work in loose collaboration with a partner, for mutual support in planning, programming, and carrying out the physics studies. A new partner will be assigned for each project. This has been found to be necessary both to keep the working relationship fresh and to ensure full exchange of the lessons and ideas learned across the course of the semester.
>
> (4) You will turn in a project report will consist of the code, results (output and figures), and a report in the form of a paper exploring the computational issues, validation procedure, and

physical insights. Part of our group discussions will be devoted to comparing our results, determining what more might be needed in the way of validation and numerical or physical exploration, and planning the contents of this paper.

You will be expected to develop code collaboratively. *This will require writing code which is structured sufficiently well to allow multiple people to contribute at the same time.* However, you will write your project reports separately, and *your project reports will be separately evaluated on the quality of work and understanding they communicate*.

Grading

Grades will be assigned based on the following aspects:

Reading Quizzes (10%) – completion of required readings before classes;

Discussion/Participation (15%) – active and engaged particpation during in-class discussion and development time;

Projects (25% each) – project reports and code, with rubric distributed later.

Topics

July 25, 2019