PHYSICS 573: NUMERICAL METHODS IN PHYSICS
PHYSICS 643: COMPUTATIONAL PHYSICS
Spring 2024

Instructor Information

Instructor: Dr. Yang Zhang, Assistant Professor in Physics & Astronomy, and EECS
Office: 217A Nielsen building
Email: yangzhang@utk.edu , or via the Canvas message system
Zoom link: https://tennessee.zoom.us/j/87524909643

Communication: The majority of classroom communication will be conducted via Canvas for this class. To ensure a prompt response, please follow the email policy:

- Please put "**PHYS 643**" or "**PHYS 573**" in the subject line of all course-related emails. This practice helps identify course-related emails.
- Before emailing, please ensure that the information is not already provided in the course syllabus or on Canvas.

General Course Information

Lecture Hours	08:10 – 09:25 am Tue/Thu
Location	NIELSEN 304
Office Hours	09:45 –10:45 am Tuesday, Nielsen 217A
Laboratory Hours	Thursday by schedule (Programing practice)
TA	Louis Primeau lprimeau@vols.utk.edu

General Course Description

Physics 573/643 is a 3-credit-hour graduate-level physics course with programming. This course covers following topics:

- Introduction to scientific and high performance computing (Linux and GPU accelerators);
- Basic numerical programming/plotting with Python and scientific packages;
- Numerical differentiation/integration;
- Computational modelling of physical problems including equation of motion (N-body problem, Chaos);
- Monte Carlo methods;
- Eigenvalue problem, electron wave equation, Hartree-Fock and mean field method,
- Exact diagonalization,
- Basic machine learning.

After successfully completing this course, students should be able to:
1) Use computational techniques to solve and explore topics addressed through the undergraduate physics program;
2) Develop and use elementary computational methods to solve physics problems;
3) Explain computational approach and methods used for selected homework to class;
4) Develop an individual semester-long computational physics project with advice of instructor;
5) Present computational physics project to class at end of the semester with write-up.

Prerequisites

Familiarity with classical mechanics and quantum mechanics, and calculus concepts, as well as concepts in vector algebra. A background in Physics up to the level of Phys 521-531, or equivalent, is highly recommended and is probably necessary for success in the course. Basic Python programming skills are required.

Textbooks

Recommended books for the course:
"Computational Physics", Nicholas J. Giordano and Hisao Nakanishi
http://www.physics.purdue.edu/~hisao/book/
"Computational Physics 2nd Edition by Jos Thijssen"
https://doi.org/10.1017/CBO9781139171397

Course Format

The course consists of two lectures per week, the Thursday class will be more towards hands-on programming. We will proceed using a mix of traditional lecturing, and problem-solving demonstrations/active-learning exercises.

Announcements, Lecture Notes, Course Updates

This syllabus and other important information and announcements will be posted on Canvas, as well as copies of the slides used in the lecture. Your grades will be posted in the Canvas Gradebook, and your grades will be available for only you to see.

Class Schedule

The following is a class schedule along with lecture topics, assignments, etc. This is a tentative schedule and might differ as our class speeds. We will be performing our computation locally and also remotely on UT's supercomputer, ISAAC.

The main focus of this course will be to provide an introduction to modern numerical techniques with the goal of either simulating or solving real physical systems. We will study examples from classical mechanics, electricity and magnetism, chaos, and statistical mechanics with an emphasis on the graphical representation of results. We will use Python as the main programming language with libraries such as Matplotlib, Scipy, and Numpy being employed where appropriate. A rough breakdown of topics includes:

- Introduction to the Phys 573/643 (Class 1)
- Introduction to Linux, scientific and high performance computing. (Class 2)
- Introduction to the Python and Linux system tools for HPC. (Class 3-5)
- Data structure, data storage, errors, and uncertainties in computation. (Class 6-7)
- Finite difference methods (dissipation in classical mechanics, chaos, equation of motion, and N-body problems) (Class 8-11)
- Derivative and Integration (Exact and Monte Carlo methods for higher dimensional integrals) (Class 12-13)
- Classical many-body problem via Monte Carlo and molecular dynamics. (Class 14-17)
- Schrödinger equation, electron wavefunction and Hartree–Fock method. (Class 18-21)
- Exact diagonalization, Lanczos algorithm. (Class 22-23)
- Interpolation, splines, and Fourier Transforms (curve fitting and analysis of experimental or simulation data) (Class 24-25)
- Introduction to Machine Learning (Class 26-27)

Class sessions:

1.	Jan. 23	Introduction to Phys 573/643	
2.	Jan. 25	Introduction to Linux and HPC	
3.	Jan. 30	Introduction to Python	
4.	Feb. 1	Hands on practice	
5.	Feb. 6		
6.	Feb. 8		1st HW due
7.	Feb. 13		
8.	Feb. 15		
9.	Feb. 20		
10.	Feb. 22		2nd HW due
11.	Feb. 27	Term project introduction	
12.	**Feb. 29**		
13.	Mar. 5		
14.	Mar. 7	2-min presentation on proposal	Project proposal
	Mar. 12,14	Spring break	

16.	Mar. 19		
17.	Mar. 21		4th HW due
18.	Mar. 26		
19.	Mar. 28	No Class Day	
20.	Apr. 2		
21.	Apr. 4		5th HW due
22.	Apr. 9		
23.	Apr. 11		
24.	Apr. 16		
25.	Apr. 18		6th HW due
26.	Apr. 23	In-class Exam	
27.	Apr. 25		
27.	Apr. 30	Term Project presentation	
28.	**May. 2**	**Term Project presentation**	

Grading & Evaluation

The semester Grade will be based on **Weighted Averages** of the homework assignments, in-class participation, the laboratory practice, mid-term exams, and the final project as follows:

Bi-weekly assignments	**40%**
One hour In-Class Exam	**15%**
Term Project	**30%**
Final Project Oral presentation	**10%**
Discussion participation	**10%**

Homework Assignment:

You will be assigned programming homework every two weeks. Each problem set will generally be available online on Tuesday before class and will be due at 11:00 pm as shown in the course schedule. **Due dates for problem sets are firm. Please note: No extensions or make-up problem sets will be given.**

The **In-Class Exams** will be in 60-min, for solving 3 problems with programming.

The Term Project is mandatory. Missing the Term project presentation is very serious and may result in the course's failure. The final project **presentation (20 mins)** will be given on Apr. 30 and May. 2 in class.

Conversion to Letter Grades

A	85 – 105
B	75 - 85
C	60 - 75
D	50 - 60
F	0 - 50

Questions and Appeals

I encourage you to ask questions during the lecture or/and talk to me during my office hours (Tuesday or by appointment – just ask after class) about the subject. You can discuss with me and/or complain to me about the grading of a given assignment, In-class Exams, or Final project. Any appeal will be entertained if it is raised no later than one week after the date on which the graded Exam/HW is made available for return to the class. After this "appeal period" of one week, exam grades will be considered final and will not be altered.

Your Feedback/Suggestions on the course

You are encouraged to provide feedback on any aspect of the course all through the semester using any communication method you prefer. You will also have an opportunity to give feedback at the end of the semester through the Course Evaluation System. Your feedback is critical in improving the course!

For students with disabilities

The University of Tennessee, Knoxville, is committed to providing an inclusive learning environment for all students. If you anticipate or experience a barrier in this course due to a chronic health condition, a learning, hearing, neurological, mental health, vision, physical, or other kinds of disability, or a temporary injury, you are encouraged to contact Student Disability Services (SDS) at 865-974-6087 or sds@utk.edu. An SDS Coordinator will meet with you to develop a plan to ensure you have equitable access to this course. If you are already registered with SDS, please contact your instructor to discuss implementing the accommodations included in your course access letter.

Academic Honesty

All work submitted by a student is expected to represent his/her own work. Students are expected to enter their own homework without assistance from others. Students are expected to perform all work in conformance with the University policies regarding Academic Honesty.