Physics 129AL: Spring 2025
Instructor: Zihang Wang (zihangwang AT ucsb.edu)
Office: Broida 6304

Introduction to Computational Physics

SYLLABUS

This course is a **physics orientated** survey of basic concepts in modern computation, and the following outline is subject to change:

- **Git/Github, Docker, and Bash/Bash script**
- **Basics in Differential Geometry** matrix, tensors, metric, dual space, generalized coordinate transformation, vector, covector, covariant derivative, parallel transport, geodesic, surface derivatives, first/second fundamental forms, intrinsic/extrinsic curvatures. Numerical Mesh generation, Convex hull, Delaunay triangulation, lifting maps.
- **Turing Machine and Computational complexity** Deterministic and Non-deterministic Turing machines, L,NL,P,NP problems, decision problem, function problems: such as counting problem, search problem, optimization problem, traveling salesman problem, quantum parallelism.
- **Basics in Matrix Theory** Gaussian and Gauss-Jacobi elimination, backsubstitution, pivoting, LU decomposition, Cholesky decomposition, QR decomposition, sparse matrix linear algebra, QR decomposition and tridiagonal forms, diagonalization of a symmetric and non-symmetric matrix, principal axes and covariance matrix, singular value decomposition (SVD), normal equations, principal component analysis (PCA) and dimensionality reduction, independent component analysis (ICA)
- **Common stochastic processes and statistical distributions in physics**, Concepts in probability and distributions, Bayesian inference and Frequentist statistics. random walk, Markov chain, geometric distribution, central limit theorem, Bernoulli process, binomial process, Poisson process, Lorentz (Cauchy) distribution. Bose–Einstein statistics (Bose-Einstein Condensation), Fermi–Dirac statistics, Maxwell–Boltzmann statistics.
- **Common distribution sampling techniques in physics**, Monte Carlo methods, stochastic sampling, inverse transform sampling, rejection sampling, gibbs sampling, Metropolis–Hastings algorithm, simulated annealing, legendre transform.
- **Foundations in neural network and artificial intelligence (AI)**, Pytorch, backpropagation, activation, feed-forward neural network, convolutional neural network, recurrent neural networks, generative adversarial network, Transformer, Autoencoder neural networks.
- **Common computation techniques in physics**, discrete Fourier transform, numerical integration and differentiation, Gaussian quadrature, orthogonal (Legendre) polynomials, implicit and explicit iterative methods for differential equations, Runge–Kutta methods, Leapfrog, symplectic integrator, (stochastic) gradient descent, explicit/implicit regularization.
- **Applications in physics**, Electrostatics, Diffusion, Brownian motion, driven system, hydrodynamics, phase transitions, molecular dynamics, *ab initio* approaches to electronic structure, quantum state (density matrix) evolution, surface code, quantum master equation, numerical renormalization group.

- **Software in Modern Computational Physics**, Quantum Espresso and LAMMPS

Students are required to have backgrounds in classical mechanics, quantum mechanics, and statistical mechanics. Here are some examples:

- Action, Euler–Lagrange equations, phase space, Hamiltonian mechanics.
- Wavefunction, eigenstate, density matrix, commutator, real space, reciprocal space, angular momentum, spherical harmonics.
- Partition function, free energy, (micro, grand) canonical ensemble, thermodynamic limit, correlation function.

In addition, students are required to have knowledge in Python:

- Python basic syntax, list, dictionary, functions, data structures, read/write, functions, objects.
- knowledge in numby and matplotlib, scipy

EQUIPMENT REQUIREMENTS

Following the physics department guidelines, students are recommended to purchase a specific raspberry pi 4 kit, but in this course, you are not required to have one. Students are expected to have **a Linux kernel** installed to preform necessary tasks. This will be done via **Docker**. In the remaining of the course, we will evaluate your work on **Github**. We will discuss the requirements in details during the first lecture.

READINGS

- The Linux Command Line, Fourth Internet Edition William E. Shotts, Jr.
- A Student's Guide to Python for Physical Modeling: Updated Edition J.M Kinder and P. Nelson
- Numerical Recipes, by Press. W. et al.
- Information Theory, Inference and Learning Algorithms, by David MacKay
- An Introduction to Statistical Learning, by James G. etal,
- A Survey of Computational Physics by Landau, R., Paez, M-J., Bordeianu, C.
- Think Bayes 2 by Allen B. Downey
- Computational Physics by Mark Newman
- From Python to Numpy by N. P. Rougier

I will assign readings or provide lecture notes based on different topics.

PROBLEMS SETS, SECTION PROJECTS AND FINAL PROJECT

There will be approximately **four (4) problem sets** , **six (6) section assignments**, and **one (1) final project**. Late homework will not be accepted except at the discretion of the instructor. Most of the problem sets will be posted in the PDF format on the course website, and your lowest problem set score will be dropped. Please check instructor announcements frequently to avoid delays.

GROUP WORKS

Students are free to form their own groups, and consider permanent by the end of Week 3. **Section work, problem sets, and the final project should be completed in groups.**

Each group must consist of 4 to 5 students. The names of all group members must be submitted to the TA via email by the end of Week 3. Each group will complete weekly section projects together. Only one submission per group is required. Each submission must include a detailed contribution report, specifying:

- Individual contributions (e.g., coding, mathematical derivation, data analysis, etc.).
- Percentage contribution of each student (must sum up to 100% per group).
- Every student must contribute to hands-on work. Contributions limited to "organizing" or "brainstorming" are not acceptable.

FINAL PROJECT AND PRESENTATION

Each group must submit:

- A written report detailing the project.
- A Git repository containing all relevant code and documentation.
- Each group will present their project to the class. Other groups will evaluate the presentation based on criteria such as innovation, correctness, completeness, and clarity. Each group will assign a score to the presenting group. The total score from all other groups will serve as the presentation score for the final project.

FINAL EXAM

No final exam during the finals week.

LECTURES, DISCUSSION SECTIONS, AND OFFICE HOURS

Lecture: TR2:00 PM-3:15 PM, Location, Phelps 3505
Sections: M,W, 3:30 PM-4:45 PM, Location, SSMS, 1304
Office hours, TBA

GRADING

With possible changes, a letter grade will be assigned based on the following weighted average:

- **Lecture participation** 10%
- **Section participation** 10%
- **Problems Sets**: 30%
- **Section Projects**: 30%
- **Final Project**: 20%