PHY 604: Final project examples

The idea behind the final project is to go beyond what we discussed in class and either solve a different physics problem with one (or multiple) of the numerical methods discussed in class, or implement a method/algorithm beyond what was discussed in class. The project will involve turning in your code, a short writeup, and presenting your results. Details and due dates are provided in the accompanying document `final_project_rubric.pdf`.

The topic is up to you, below are some that you can either choose, or use as inspiration. Note that they are relatively open ended; you should make sure that your project is specific enough to complete in the time allotted. I have also included some references to start, but you may need to use others (let me know if you need to borrow any of the books that are referenced).

1. *Lanczos method for matrix diagonalization*: One of the most powerful mnethods in large-scale matrix computing is the Lanczos method, which is an iterative scheme suitable for large, especially sparse, matrices. The advantage of the method is most noticable when we only require a few eigenvalues and eigenvectors. Implement this method and demonstrate that it can be use to diagonalize large matrices. (More information can be found in, e.g., Section 5.9 of Pang, or Section 11.3.2 of "Computational methods for physics," by Joel Franklin).

2. *Conjugate gradient*: A method of deterministic optimization that works well for, e.g., very multidimensional problems is the conjugate gradient method. Implement this method and test in on several minimization problems. (More information can be found in "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain," by Jonathan Richard Shewchuk, linked in the Lecture 10 slides, and Section I.3 in "Basic concepts in computational physics," by Stickler and Schachinger.)

3. *Multigrid methods for differential equations*: Multigrid methods are a powerful set of tools for accelerating relaxation methods. Implement this method and test on some simple boundary-value problems on spatial domains (e.g., solving the Poisson equation). (More information can be found here, and in specialized texts such as "An introduction to multigrid methods," by Wesseling or "Multigrid methods and applications," by Hackbush).

4. *Monte Carlo for the Potts Model*: The Potts model is a classic model in statistical physics. It is a generalization of the Ising model to spin realizations that can take integer values $\sigma_i = 1, 2, \ldots, q$ (the Ising model is the case of $q = 2$). Write a program to solve the Potts model for arbitrary q, and investigate observables like the energy, magnetization, and specific heat. (More information can be found in Section 18.3 of "Basic concepts in computational physics," by Stickler and Schachinger.)

5. *Wavelet analysis*: One shortcoming of Fourier analysis is that it does not contain time-resolved information about the frequencies present in a signal. Therefore, it is not ideal for signals that are not stationary in time, i.e., whose form changes with time. Wavelet analysis is intended to address this shortcoming by expanding a signal in a complete set of functions (wavelets), each of which oscillates for a finite period of time and each of which is centered at a different time. Implement the wavelet transformation and apply it to nonstationary signals. (More information can be found in Chapter 11 of "Computational Physics," by Landau, José Páez, and Bordeianu and Sections 6.6 and 6.7 of Pang).

6. *The Numerov method*: Investigate the Numerov method for Schrödinger's equation—this is a special ODE integrator that is designed to work on second-order ODEs directly. Write a solver uses shooting to find bound states and eigenvalues for a general potential, and test it on various potentials (More information can be found in Section 4.9 of Pang).

7. *Molecular dynamics*: Molecular dynamics (MD) is a powerful simulation technique for studying the physical and chemical properties of solids, liquids, amorphous materials, and biological molecules.

Implement code to do MD simulations using Lennard-Jones intermolecular potentials and the Velocity-Verlet algorithm. (More information can be found in Chapter 16 of "Computational Physics," by Landau, José Páez, and Bordeianu, Chapter 8 of Pang, and Chapter 7 of "Basic concepts in computational physics," by Stickler and Schachinger)

8. *Chaos*: Explore a chaotic ODE system (e.g., the double pendulum, damped driven pendulum). Vary parameters to map out different regions, showing chaotic behavior, period doubling, etc. (For more information see Chapter 13 of "Computational methods for physics," by Joel Franklin and Chapter 12 of "Computational Physics," by Landau, José Páez, and Bordeianu)
9. *SIAM top 10 algorithms of the 20th century*: For some additional ideas, see: this list.