PHY 250L – Spring 2017
Final Projects

This document lists the assignments for the final computing project in PHY 250. Projects are pre-assigned to teams of (mostly) two students. If you don't like your project, feel free to propose another, but it must be approved before you proceed!

Projects will be completed by the last day of lab, 5.9. During this last lab meeting, you will give a brief semi-informal presentation to the class and departmental faculty, describing the physical/mathematical problem which you investigated and an overview of your code. We will talk more about final presentations as their date approaches.

PLEASE begin working on these problems SOON! You will have a reduced homework load over the next few weeks, anticipating that you'll spend most of your time on your projects.

---

1. **Lottie, Pat**

   In this problem, you will simulate the diffusion of heat through a 2-dimensional metal plate. In such situations, the temperature $T(x, y, t)$ at time t at position (x, y) is given by the *heat equation*:

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0 \tag{1}$$

$$\implies \frac{\partial T}{\partial t} - \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 \tag{2}$$

   In your problem, $\alpha = 111$ mm^2/s is the thermal diffusivity of copper.[1] The plate is a square with edge length 0.5 m. Think of your plate as lying in the first quadrant with $x, y \in [0, 0.5]$.

   [1] α tells how quickly heat travels through a given material.

   The plate's temperature is fixed at its center and edges: 600 K and 300 K, respectively. At $t = 0$, the interior of the plate is also at 300 K, but the temperature in this region will change as t increases, in accordance with the heat equation. Your job is to simulate the diffusion of heat throughout the plate, and produce plots of the temperature vs (x, y) at intervals of 25 s. Save these images for later viewing.

   The first step in this problem is to think about dividing the plate up into small square chunks, given some dx and dy. You can then create a 2-d array that represents the temperature values at each (x, y) location. Begin by choosing $dx = dy = 1$ mm, but expect to decrease this once your simulation is well understood. You will then evolve the system forward in time by choosing some dt and updating the temperatures at each (x, y). Begin with $dt = 0.1$ s. Your simulation should run until $T(x, y, t)$ reaches a "stable-enough" configuration.

   It's not necessary to produce a "movie" of this evolution, but it sure would be cool!

---

2. **Matt, Jeremy, Ashley**

   In this problem, you will simulate the physics of the 1909 Geiger/Marsden Rutherford scattering experiment that suggested the existence of the atomic nucleus.

You will simulate the trajectory of a beam of α particles from decay of ^{210}Po incident on an Au atom. The α particles can be considered to travel along the z-axis toward the Au atom located at the origin. The initial positions of the α particles should be randomly generated within a distance of a_0 from the negative z-axis, where a_0 is a rough estimate of the radius of the gold **atom**.

Note that the force exerted on an α particle by the atom is *zero* if the α is outside of the electron cloud (*i.e.*, farther than a_0 from the nucleus)!

Your work should demonstrate three effects:

- plots of roughly 100 sample α particle trajectories (on the same axes)
- a histogram of the polar angles of the out-going α particles for at least 2000 α particles. Recall that the plum pudding model predicted no back-scatter, contrary to what Geiger and Marsden observed.
- a scatter plot of out-going (scattering) angle versus impact parameter[2] for at least 2000 α particles.

[2] Look it up.

Even though you are interested in the out-going polar angles, you will want to code your simulation in terms of Cartesian coordinates... trust me.

You might want to prototype this in VPython first...

---

3. **Akira, Zach**

In this problem, you'll simulate the vibrations in a membrane stretched over a square armature (*i.e.*, a square drum). For a two-dimensional medium such as this, the height (or displacement from zero height) of the membrane, $z(x, y, t)$ is dictated by the *wave equation*:

$$v^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) = \frac{\partial^2 z}{\partial t^2} + \gamma \frac{\partial z}{\partial t}, \tag{3}$$

where v is the *phase velocity*, *i.e.* the speed at which vibrations travel on the membrane, and γ is a damping coefficient.

Assume that your membrane is stretched over an armature that forms a square in the first quadrant, $x, y \in [0, 1]$. (It's a **big** drum.) The rim of the drum fixes $z(x, y, t) = 0$ for all points on the rim at all times. Assume that $v = 100$ m/s, $\gamma = 0.02$, and that the initial configuration of the drum head is

$$z(x, y, t) = \begin{cases} ((-2.0 \text{ cm})|0.5 - x| + 0.01)\ \ y, & \text{if } y \le 0.75 \\ ((-2.0 \text{ cm})|0.5 - x| + 0.01)\ \ (3 - 3y)\,, & \text{if } y > 0.75 \end{cases} \tag{4}$$

See the instructions in the first problem for how to proceed with this simulation. You will choose $dt = 0.00001$ s and plot the configuration of the drum head every 0.05 s.

---

4. **Jiaoyi, Theo**

In this problem, you will investigate a system of famous differential equations: the *Lorenz Attractor*. In the mid-1960s Edward Lorenz developed the following set of equations in attempting to describe atmospheric/meteorological effects:

$$\dot{x} = \sigma(y - x) \tag{5}$$
$$\dot{y} = x(\rho - z) - y \tag{6}$$
$$\dot{z} = xy - \beta z, \tag{7}$$

where σ, ρ, and β are constants. (Though Lorenz developed these equations for three quantities representing atmospheric observables, we'll consider them as functions of spatial coördinates, x, y, and z.) Recall that in the "dot notation", $\dot{x}$ is shorthand for dx/dt.

This set of equations was one of the first systems that was observed to demonstrate *chaotic behavior*, meaning that the behavior of the system depends *very* strongly on its initial conditions. The trajectories given by the Lorenz Equations are deterministic, but it is very difficult to approximate the future state of the system given a set of initial conditions. Said another way, very small changes to the initial conditions can lead to *wildly* different behavior of the system.

Lorenz found that this system of equations exhibits chaotic behavior for values *near* the following:

$$\sigma = 10, \quad \beta = 8/3, \quad \rho = 28 \tag{8}$$

Your job is to investigate the behavior of a particle whose dynamics are described by the Lorenz equations. First, generate the trajectory with initial position $\vec{r}_0 = (1, 1, 1) \equiv \vec{1}$.[3] Use a timestep of $dt = 0.005$, and write to file the position of the particle at 0.1-s intervals for a duration of 100 s. Call this trajectory the "standard" trajectory.

[3] Do *not* use the sample code that is available on Wikipedia (or elsewhere). Program so that the three components are updated *explicitly*.

Then generate trajectories with different initial conditions. Choose 50 different initial positions with components in the range $[0.95, 1.05]$, and rerun the simulation, writing files similar to the one that you constructed for the unit $\vec{r}_i$. Call these trajectories the "modified" trajectories.

Write a piece of code that reads in one of these trajectory files and plots it in 3d.

Now you'll compare the difference between the modified trajectories with the standard trajectory as a function of time:

$$\delta(t) = \left(\vec{r}_s(t) - \vec{r}_m(t)\right)^2 \tag{9}$$

Plot the $\delta(t)$ versus t on five separate sets of axes (ten per axes), grouped by how far the initial position is from $\vec{1}$.