Projects for PH 430 Numerical Methods and Scientific Computation

Consult the faculty for more information on these projects

For each project you need to write up a short report and give a 10 minute presentation.

Problem 1: Billiards and Chaos

Write a program that tracks the trajectory of a single ball rolling on a frictionless flat table with walls at its edges. The ball only changes its velocity through collisions with the walls; in which case the ball is reflected specularly – its speed is conserved and the angle of incidence equals the angle of reflection. Contrast the trajectories where the walls have a) a circular shape with radius \( r = 1 \), and b) a stadium shape with two semi-circles connected with straight walls with the semi-circle radii a distance \( d = 2\alpha r \) appart. Make Poincaré sections of phase space plots with dots depicting \( v_x \) versus \( x \) whenever \( y = 0 \). Do this for \( \alpha = 0, 0.001, 0.01, 0.1 \). Also plot the trajectories themselves. Consider the divergence of two trajectories with nearly identical initial conditions and estimate the Lyapunov exponent.

Problem 2: Schrodinger’s equation for two 1D Potentials (ODE)

A. Use the Verlet method to solve the Schrödinger equation for the potential \( V(x) = 2x^4 \) to obtain the lowest 8 eigen-energies and eigenfunctions. Write a C++ program that solves the Schrödinger equation numerically using the Verlet or other method. Plot the potential, wave-functions, and eigen-energies on a single graph.

B. Now repeat the problem for the potential \( V(x) = x^4 + \lambda x^2 \) varying \( \lambda \) from 1 to -4.

C. Interpret the differences between the nearly degenerate eigen states for the double-well cases.

Problem 3: Probability Problems

Write C++ programs to solve the three problems below.

a) Assume that a particular disease affects 1% of the population of country X. The people in this are tested at random for this disease and the test is 90% accurate. If a person tests positive for the disease, what is the probability that they actually have it? Write a program that computes this probability.

b) A player rolls two dice. If the sum of the two dice is 7 or 11, the player wins immediately. If the sum is 2, 3, or 12, the player loses immediately. If the game is neither one or lost on the first throw (i.e. if the sum is 4, 5, 6, 8, 9, or 10) the player rolls the dice again until she either wins by repeating her initial number or loses by rolling a 7. Write a program to determine the probability that the player wins the game.

c) Simulate a blackjack game where the bet on each hand is fixed to be one dollar and the dealer stays at 17 or above (counting the ace as 1). Average over many games and compare the average winnings when the player stays at 16, 17 and 18 respectively.

d) Then repeat the analysis on a deck with 2, 4 and eighth picture cards removed.
Problem 4: Ant in a Maze

In this problem you will simulate how far an ant moves away from its starting point in a maze. The maze is modeled by a two dimensional $N \times N$ lattice of points, each of which is located by indices $i, j$. Each point on the lattice is either “open” with probability, $p$, or “blocked” (with probability, $1 - p$). At time, $t = 0$, the ant starts on one of the open points and at each time step tries to move randomly to one of its four neighboring points. If that point is open, the ant moves to the new location. If the point is blocked, the ant stays where it is. In both cases, time is incremented by one step after each attempt. For a particular choice of $p$, you are to calculate the distance, $R$, between the ant’s starting point and its end point after a certain time, $t$. Repeat this simulation many times for each $p$, each time giving the ant a randomly chosen open site. Now average your values of $R$ and produce a Log-Log plots of $R$ vs. $t$ for several values of $p$. Test for power-law scaling and determine the critical exponents where appropriate. Among the values of $p$ selected, you should use $p = 0.2, p = 0.4, p = 0.6, p = 0.8$, and $p = 1$.

Problem 5: Avalanches and Self-Organized Criticality

Develop a simulation of an undirected nearest-neighbour sandpile model on a two-dimensional square $N \times N$ lattice. First let this simulation of non-equilibrium dynamics evolve to a steady state. Then start developing statistics of the distribution of sizes of sand avalanches $P(s, N)$ resulting from the addition of a single grain of sand, as well as the distribution of the amount of sand that subsequently falls off the two edges of the cylinder, $E(w, N)$. Consult the instructor on your progress.

1. Plot $P(s, N)$ vs. $s$ for several increasing $N$, show that the steady state of the sandpile is critical and determine the relevant power law exponent.
2. Next plot $E(w, N)$ vs. $w$ for several increasing $N$, and determine the relevant power law exponent.

Repeat the above analysis for a generalisation of the model where sand is not conserved but lost with a 2% probability at each toppling event. Interpret the results and explain why the results differ in this case.

Problem 6: Diffusion-Limited Aggregation

Write a program to simulate the growth of a snowflake through Diffusion-Limited Aggregation on a two-dimensional square lattice. Visualize the results graphically and determine the fractal dimension.

Problem 7: Self-Avoiding Random Walk

Develop a simulation for the self-avoiding random walk on a two-dimensional square $N \times N$ lattice. Calculate the average distance $R$ of the walker after $I$ steps, $R(I, N)$ as a function of system size and number of steps, averaging over many trials.

1. Plot $R(I, N)$ vs. $I$ for several increasing $N$.
2. Get an approximate value for $R(I, \infty)$ for number of steps where your large $N$ data does not show $N$-dependence. Plot $R(I, \infty)$ and extract the exponent $\nu$.
3. Understand the concept of finite-size scaling and do your best to collapse all of your $R(I, N)$ data on a single scaling function.
Problem 8: Scaling and critical coupling for synchronous order in pulse-coupled oscillators

Solve for \( N \) interacting integrate-and-fire neurons described by the set of ODE’s

\[
\tau_m \frac{dv_i}{dt} = -(v_i - E_L) + R I_i + \mathcal{V}(t) \quad i = 1, 2, \ldots N
\]

plus threshold rule that \( v_i(t) \) gets reset to \( v_{\text{reset}} = -70 \text{ mV} \) after it reaches \( v_{\text{thresh}} = -50 \text{ mV} \). Use \( E_L = v_{\text{reset}} \), \( \tau_m = 20 \text{ ms} \) for each neuron and take \( R I_i \) from some distribution. The neurons interact by producing axonal currents \( A_j(t) \) that are injected as additional input into all the neurons (including itself). Take \( \mathcal{V}(t) = (K/N) \sum_{j=1}^{N} A_j(t) \) where \( K \) is the synaptic coupling strength and the current \( A_j(t) \) is comprised of a sequence of pulses following each action potential (threshold condition) of neuron \( j \). So if neuron \( j \) fires at times \( f^{(j)}_1, f^{(j)}_2, \ldots, f^{(j)}_M \) then \( A_j(t) = \sum_{l=1}^{M} f(t - f^{(l)}_j) \) where \( f(t) \) is the pulse shape.

Map the dynamic voltage variable \( v_{\text{reset}} \leq v_i(t) \leq v_{\text{thresh}} \) to a dynamic phase variable \( 0 \leq \phi_i(t) \leq 2\pi \). Then use this mapping to determine the order parameter

\[
M(t) = \frac{1}{N} \sum_{i=1}^{N} e^{i\phi_i(t)}.
\]

Part A: Choose the initial \( v_i(0) \) to be random and use an exponential pulse shape \( f(t) = (1/\tau_s) e^{-t/\tau_s} \theta(t) \) with \( \tau_s = 5 \text{ ms} \) and \( \theta(t) \) the Heavyside step function. Take \( N \) sufficiently large. Choose the distribution of constant input currents \( R I_i \) to be a delta-function: \( \mathcal{P}(RI) = \delta(RI - R I_{35}) \) – i.e., all neurons have the same natural period, namely 35 ms.

Determine the average asymptotic value of \( R = |M| \) for different values of the synaptic coupling constant \( K \) and plot \( R \) vs. \( K \) on linear and log-log plots.

Determine the parameters in the scaling form

\[
R \propto |K - K_c|^\beta.
\]

Part B: Repeat the same analysis but for a gaussian distribution of input currents

\[
\mathcal{P}(RI) = \frac{1}{\sqrt{2\pi W}} e^{- (RI - R I_{35})^2 / 2W^2}
\]

with \( W = 1.5RI_{35} \). (Note keep all input currents above the threshold of 20 mV.)

Project 9: Generalised Hopfield Neural Network with grayscale pictures defining local energy minima

In class, we studied a Hopfield Neural Network Model that used a 2D Ising Model with appropriate interaction \( J_{ij} \) which minimised symbols in a library to model associative memory. Generalise the method to the case where the microscopic spin variable can have one of \( G = 32 \) values (instead of two) and where the library images are appropriate gray-scaled picture files – perhaps a few faces.

You will need to generalise the Ising hamiltonian to include an external field, and possibly a Lagrange multiplier to help the Monte Carlo Simulation find the correct local minimum.

Projects 10-13: to be coordinated with the instructor

Project 10: develop a code for the “Game of Life” simulation

Project 11: fractal generation and dimensionality

Project 12: molecular dynamics

Project 13: protein folding