ASTRONOMY 598: MONTE CARLO METHODS HOMEWORK 7

DAVID FLEMING

Contents

README 1
Running on Hyak 1
Problem 1 1
1a 2
1b 2
1c 2
1d 2

README

This directory contains the code that answers question from homework 7 while this document provides additional content for the same questions. Specifically, `run_hw7.py` implements a self-avoiding random walk and plots the results of various runs. The script `run_hw7.py` generates the accompanying figures as well. To run, enter `python run_hw7.py`.

Running on Hyak

To run the code on Hyak, follow the instructions given below.

1) Create an interactive session by entering `qsub -I -l walltime=hr:min:sec` where hr = 03 is a safe amount of time

2) Find your favorite python distribution (2.7+ for this code) using `module avail`

3) Load the python distribution via `module load (name of package found using module avail)`. I recommend loading `anaconda_2.4`.

4) Run the script by typing `python run_hw7.py`

Problem 1

The code and figures for all parts are given in the accompanying file. Note: entering `python run_hw7.py` in the terminal will generate all the plots for this homework.

1a. My implementation for the self-avoiding random walk is included with comments in the file `run_hw7.py`. Instead of running the random walk for N = 10, 20, 30, and 40 steps, I chose 5, 10, 15, 20, 25, 30 as my random walker would have very, very few acceptances for N larger than 30.

1b. See the accompanying plot 1b.png. It appears that $< s^2 >$ for a self-avoiding random walk increases with N more steeply than for a typical random walk.

1c. See the attached 1c.png for the plot. When I fit log$< s^2 >$ as a function of logN, I found that the slope was around $1.2 - 1.3$ depending on the run. Since log$< s^2 >$ as a function of logN is about 1 for the typical random walk, this fit confirms that the self-avoiding random walk travels farther on average when it is able to successfully avoid itself.

1d. See the attached 1d.png for the plot. The acceptance ratio rapidly decreases with increasing step size. This makes sense because the more a random walker has to travel, the more likely it is to run into itself.