Physics 114 **Statistical Mechanics** **Spring 2021**

Week 4 Reading and Problem Assignment

Overview:

This week we study the fundamentals of *probability theory.* We go over the definition of a probability distribution, both for *discrete* and *continuous random variables.* We use a computer to generate random numbers, and either use pre-written G&T codes or write our own, with algorithms involving *random numbers.* We practice counting states for systems of interest, so that we can derive probability distributions - like the ubiquitous *binomial distribution.* We learn to calculate the *moments* of a probability distribution, with especial emphasis on the concepts of *mean* and *variance.* For discrete distributions involving a large number N, we will find it useful to use *Stirling's approximation.* We (finally!) encounter a definition of entropy based on the probability of observing a macrostate. Section 3.4.1 of G&T talks about entropy in terms of *uncertainty.* A recommended reading by Pratt talks about *ignorance.* With these in mind, we are meant to believe that $S(\Omega) \propto ln\Omega$ is an excellent definition of how uncertain/ignorant we are about the results of a measurement, if all measurement results, Ω, are equally probable. The more general case, if a measurement result has a probability of P_i, would be $S = -k\ P_i\ lnP_i$.

Spoiler alert: Next week, we will continue to think about in entropy in a statistical way. Section 14.8 of B&B will call $S = -k\ P_i\ lnP_i$ the *Gibbs entropy* of a thermodynamic system. B&B Ch. 15 defines the *Shannon entropy*, which applies beyond thermodynamics, to fields like cryptography, data compression and quantum mechanics.

Suggested Reading:

G&T Sections

- Ch. 3.1-1.7 and 3.11.1
  Notes:
  - We will save Bayes Theorem, section 3.4.2 for next week
  - It's up to you whether you want to read the rather technical section 3.11.2, a proof of the Central Limit Theorem. It introduces key concepts of probability theory: cumulants and characteristic functions.
- Section 4.1

B&B sections

- Section 1.4

- Ch. 3
- Appendices C.1 , C.2, C.3 and C. 13

Optional Reading:

- There is a short reading by Pratt on "ignorance" and Lagrange's undetermined multipliers Resources" area of our Moodle site.
- Calculating Lagrange's undetermined multipliers: a Wolfram Widget: `http://www.wolframalpha.com/widgets/view.jsp?id=1451afdfe5a25b2a316377c1cd488883` There is also this demo, a Wolfram Demonstrations Project: `http://demonstrations.wolfram.com/LagrangeMultipliersInTwoDimensions/`

Warmup Problems:

1: Independent spins G&T Problem 4.1
2: The exponential distribution G&T Problem 3.39
Hint: You can check your answer to (a) b/c it is given in G&T section 3.9.

Problems to discuss in our meeting

Note: The * means that these problems are to be handed in. They are due the day after we meet.

1: I heart Random numbers ... a numerical problem
This is a problem asking you to do some amount of numerical computation. Preferably using Python ...

a) Generate a set of 200 random numbers $\{x_i\}$,where $i = 1, ..., 200$. These numbers should be uniformly distributed between 0 and 1. Please give the lines of code you used to generate these, as your answer to this problem.

b) For a uniform distribution of random numbers, $p(x)dx = dx$. In other words, $p(x) = 1$ for all x. Plot a histogram of these numbers, to see if indeed all numbers seem equally likely.

c) Find the mean and variance of your numbers, $\bar{x}$ and σ^2. Comment on whether they are close to what'd you expect from a uniform probability distribution between 0 and 1.

d) Calculate a new random number X which is the sum of all 200 numbers: $X = \Sigma_{i=1}^{200} x_i$. Do this lots of times, until you have a set of 500 numbers $\{X_j\}$ where $j = 1, 500$. Plot a histogram of these numbers. How close is your histogram to the shape of a Gaussian? Does the probability distribution of these numbers, $p(X)$, have a mean and variance that is close to what is predicted by the central limit theorem?

2: Binomial and Gaussian* B&B Problem 3.7
Hint: In Part b) B&B say it's hard to show that the binomial distribution approaches a Gaussian. G&T do show it on pages 145-146 using Stirling's approximation.

3: Basic probability:
i) **What are the rules?** G&T Problem 3.7
ii) **Likelihood of various outcomes?** G&T Problem 3.18
Note: ii) assumes gender is binary :-{
iii) **Expectation values:** B&B Problem 3.5 parts (a), (c), and (e)

4*: How big is that pond, and how many fish are in it?
i) G&T Problem 3.58
ii) G&T Problem 3.59

5: Monte Carlo Integration G&T Problem 3.60

6: Poisson distribution B&B Problem 3.3

7: Binomial distribution for magnets and gasses
i) G&T Problem 3.27
ii) G&T Problem 3.34

8*: Lagrange's undetermined multipliers G&T Problem 3.51

9: Stirling's approximation G&T Problem 3.33

10: Hands-on with simulations: Since you now know how to generate a set of random numbers from Problem 1 do something cool with random numbers: Here are some ideas ... you only need to **choose one**.

i) Explore one of the G&T simulations we did not use yet in a problem. That is, run the code, become familiar with what it does, and be ready to share this information and the results with your colleagues in seminar. You can find these from the Jar Launcher App or the `https://www.compadre.org/stp` as shown below

ii) Write your own Python code to create a random walk in one dimension. (This would be tantamount to your own version of G&T Problem 3.36.)

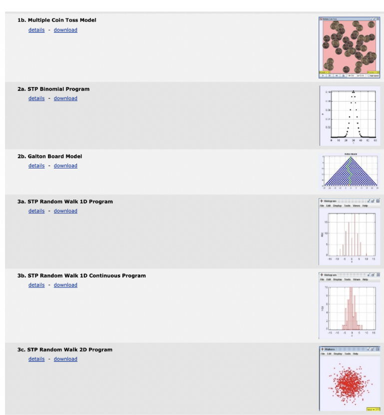
ii) Code up a random walk in two dimensions, where the particle can move one step left, right, up or down with equal probabilities, and visualize it for us with either a graph of the trajectory or, even cooler, an animation of the path taken by the walker. Calculate the square of the distance traveled from the origin, $< r^2(N) >$ where N is the number of steps taken, How does this square distance depend on N?

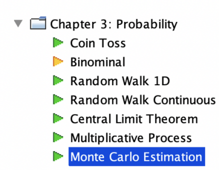
iv) Visualize many walkers at once in two or three dimensions, and you have the bones of the kinds of simulation of "approach to equilibrium" that were associated with Ch. 1.

v) Turn your uniform random numbers $\{x_i\}$ into Gaussian-distributed random numbers, $\{z_i\}$ using the Box-Muller algorithm. (See Wikipedia, or `http://mathworld.wolfram.com/Box-MullerTransformation.html`.) In a nutshell: You can transform your set of uniformly-distributed $\{x_i\}$ in pairs to generate pairs of Gaussian-distributed random numbers:

$$z_i = \sqrt{-2\ lnx_i}\ cos(2\pi x_{i+1}),\ z_i + 1 = \sqrt{-2\ lnx_i}\ sin(2\pi x_{i+1}).$$

vi) Build a probability distribution of your choice. Yes, Python's numpy library already knows a lot of distributions, but there is a cool Monte Carlo trick for doing this in a few lines of code. See for example
`https://stackoverflow.com/questions/4265988/generate-random-numbers-with-a-given-numerical-distribution.`





Jar Launcher simulations

Simulations visible at
https://www.compadre.org/stp/filingcabinet/share.cfm?UID=10986&FID=22830

G&T Codes that can be used for option i) in this problem