PHY407-Lab10

due November 29 2022

Physics Background

Hypersphere A hypersphere is the equivalent of a sphere in higher dimensions. A hypercube is the equivalent of a cube in higher dimensions.

In n dimensions, a hypersphere ("n-ball") of radius R with coordinate directions x_i, $i = 1, ...n$, is given by the points x_i such that

$$\sqrt{\sum_{i=1}^{n} x_i^2} = R. \tag{1}$$

Convince yourself that this makes sense for a 3D sphere (the one you are used to) as well as a 2D sphere (i.e., a circle). There is actually an exact expression for the volume of an n-ball of radius R:

$$V_n(R) = \frac{R^n \pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)}, \tag{2}$$

where Γ is the gamma function, i.e., an extension of the factorial operator to real numbers. For our purposes,

$$\forall x \in \mathbb{R}_{\geq 0}, \quad \Gamma(x+1) = x\Gamma(x) \quad \text{and} \quad \forall n \in \mathbb{N}^*, \quad \Gamma(n) = (n-1)!. \tag{3}$$

The volume of an n-dimensional hypercube of length L is simply L^n.

Computational Background

Random number generators There is no one single best or "go-to" python package for random number generation. We have seen a few already, and the following pages have a lot more info:
`https://numpy.org/doc/stable/reference/random/index.html`
`https://docs.python.org/3/library/random.html`

Histogramming in python For an array of values (say it's called `I`), you can make a histogram using `n` bins from minimum value `a` to maximum value `b` using the command:

```
import matplotlib.pyplot as plt
# ...
plt.hist(I, n, range=[a, b])
plt.show()
```

Questions

1. Brownian motion and diffusion limited aggregation [30%]

(a) Do exercise 10.3 on page 457 of the Newman text, but with only 5000 steps (since 1 million steps will take a long time). Rather than an animation, just make a plot of the particle's i vs t, a plot of j vs

t, and a plot of i vs j (so no need to import the `visual` package). You can start from the sample program `Brownian-start.py` provided.

Submit your code, and figures showing the trajectory (i vs t, j vs t, i vs j) of the random walk.

(b) Do exercise 10.13 part (a) on pages 499-500 of the Newman text, but with no visualization (i.e. no animation). Instead of making the animation, have the program stop once there is an anchored particle at the center of the grid (i.e. at the point where each particle starts its random walk), and then display a figure showing the positions of all the particles. You can start from the sample program `DLA-start.py`, along with the program you wrote for the previous part.

Submit your code, and the figure showing the final particle positions.

2. Volume of a 10-dimensional Hypersphere [20%]

Do exercise 10.7 on pages 471-472 of the Newman textbook. Use 1 million points to get a good estimate.
Submit your code, and the value for your integral.

3. Importance Sampling [50%]

(a) The integral

$$\int_0^1 \frac{x^{-1/2}}{1+e^x} dx \tag{4}$$

(eqn. 10.34 of the textbook) is an integral whose integrand has a divergence. Evaluate this integral using the mean value method with 10,000 sample points. Repeat this calculation 100 times.

Submit your code.

(b) Evaluate this integral using the importance sampling method (see textbook Section 10.2.3) with 10,000 sample points. Repeat this calculation 100 times.

- Use the weighting function $w(x) = x^{-1/2}$, which removes the singularity.
- You will need to determine the transformation needed to non-uniformly sample your region. The probability distribution you will be drawing from is

$$p(x) = \frac{1}{2\sqrt{x}} \tag{5}$$

Pages 458-459 of the text explain how to find the transformation for a given probability distribution, to help you understand where this $p(x)$ comes from.

Submit your code.

(c) For each of the two methods separately, make a histogram of the integral values you obtained above, using 10 bins from 0.80 to 0.88. Comment on what your histograms tell you about each method.

Submit your histograms and brief written answer.

(d) **Optional** Importance sampling also helps evaluate rapidly varying integrands even if they aren't singular. For example, consider the integral

$$\int_0^{10} \exp(-2|x-5|)\, dx. \tag{6}$$

This is a function that is sharply peaked near $x = 5$, and we can easily find its exact value. If we calculate it using points uniformly distributed on the interval $0 <= x <= 10$ we will be using a lot of points that don't contribute to the expectation value. To sample more points near $x = 5$ in order to increase the precision of the sampling method, choose an importance function that is Gaussian of the form

$$w(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2} \tag{7}$$

that will lead you to draw from the normal distribution

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-5)^2/2} \tag{8}$$

with mean of 5 and standard deviation of 1. (You can use `numpy.random.normal` to do this.)

Using the mean value and importance sampling methods: evaluate the integral using 10,000 sample points, repeat the calculation 100 times, and histogram the results.