Homework Set #5 – PHYS 6260
Prof. John Wise
Due Friday, February 14th, 11:59pm (Submit github URL to Canvas; all code on github)

- Your assignment should be uploaded as a **single Jupyter notebook** with all of the problems included.
- Please use the template notebook uploaded on Canvas and github as a starter.
- Comment your code through inline comments with # or markdown blocks, where the latter option is preferred.
- In the problem descriptions, "programs" are referring to single or multiple code blocks in a notebook.
- The materials that you are required to include are indicated at the end of each problem, next to the check symbol: ☑

---

1. **Monte Carlo Integration (45 points total):** Calculate a value for the integral

$$I = \int_0^1 \frac{x^{-1/2}}{e^x + 1}\,\mathrm{d}x, \tag{1}$$

using the importance sampling formula

$$I \simeq \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x)\,\mathrm{d}x \tag{2}$$

we covered in class with $w(x) = x^{-1/2}$ as follows.

(a) Show that the probability distribution $p(x)$ from which the sample points should be drawn is given by $p(x) = x^{-1/2}/2$ and derive a transformation formula for generating random numbers between zero and one from this distribution.

(b) Using your formula, sample $N = 10^6$ random points and hence evaluate the integral. You should get a result around 0.84.

☑ **For full credit**, you should show your work in the derivation in part(a). For part (b), include your program with comments.

2. **A random point on Earth's surface (45 points total):** Suppose you wish to choose a random point on the surface of the Earth. That is, you want to choose a value of the latitude and longitude such that every point on the planet is equally likely to be chosen. In a physics context, this is equivalent to choose a random vector direction in 3D space (something that one has to do quite often in physics calculations).

Recall that in spherical coordinates θ, ϕ (where θ is the angle from the north pole and ϕ is the azimuthal or longitudinal angle) the element of solid angle is $\sin\theta d\theta d\phi$, and the total solid angle in a whole sphere is 4π. Hence the probability of our point falling in a particular element is

$$p(\theta, \phi)d\theta d\phi = \frac{\sin\theta d\theta d\phi}{4\pi}. \tag{3}$$

We can break this up into its θ part and its ϕ part as the following:

$$p(\theta, \phi)d\theta d\phi = \frac{\sin\theta d\theta}{2} \times \frac{d\phi}{2\pi} = p(\theta)d\theta \times p(\phi)d\phi. \tag{4}$$

(a) What are the ranges of the variables θ and ϕ? Verify that the two distributions $p(\theta)$ and $p(\phi)$ are correctly normalized—they integrate to 1 over the appropriate ranges.

(b) Find the formulas for generating angles θ and ϕ drawn from the distributions $p(\theta)$ and $p(\phi)$. *Note:* The ϕ one is trivial, but the θ formula is not.

(c) Write a routine that generates a random (θ, ϕ) coordinate, using the formulae you worked out. *Hint:* In Python, the function `acos` in the `math` package returns the arc-cosine in radians of a given number.

(d) Now use that routine to generate 500 such random points, convert the angles into (x, y, z) coordinates assuming the radius of the globe is 1. Now visualize the point in 3D space in an interactive scatter plot, for example with `scatter` from the `matplotlib.pyplot` module. *Hint:* You can use the "magic" command `%matplotlib widget` as the first line in a Jupyter notebook. See the starter code in the template notebook for an example.

☑ **For full credit**, you should give an explanation for part (a), your derivation for part (b), and programs with comments for parts (c) and (d), along with an interactive 3D plot for part (d).

3. **Application question (10 points):** In a couple of paragraphs (about 250 words), discuss a system that can be modeled with Monte Carlo techniques that were not covered in the lecture examples. Provide any governing equations in this system. Which variables and processes would be modelled as random? Describe how these quantities and processes behave in a continuous medium, i.e. in "real life." You do not have to provide any code.