HW5_Solutions

February 20, 2024

1 PHYS 6260 - Homework 5 Solutions

[2]:
```
import numpy as np

from numpy import random
from math import exp, cos, sin, acos, pi

%matplotlib widget
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import rcParams
rcParams['font.size'] = 16
rcParams['figure.figsize'] = (10,8)
```

2 Problem 1: Monte Carlo Integration

Calculate a value for the integral

$I = \int_0^1 \frac{x^{-1/2}}{e^x+1} dx$

using the importance sampling formula

$I \simeq \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x) dx$

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

2.1 Part (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.

The probability distribution is given by:

$\int_0^1 x^{-1/2} dx = [2\sqrt{x}]_0^1 = 2.$

Here $p(x) = \frac{1}{2} x^{-1/2}$. By integrating over $p(x)$ from 0 to x, we have:

$z = \frac{1}{2} \int_0^x x^{-1/2} dx = \sqrt{x},$

and hence $x = z^2$. Then, we obtain:

$\int_0^1 \frac{x^{-1/2}}{e^x+1} dx \simeq \frac{2}{N} \sum_{i=1}^{N} \frac{1}{e^{z_i^2}+1}$,

where z_i is a uniformly distriubted random number between zero and one.

2.2 Part (b)

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

[3]:
```
# define number of random points
N = 1000000
# initialize s
s = 0.0

# loop over all points
# evaluate $\frac{1}{e^{z_i^2}+1}$ and sum
s = np.sum(1 / (np.exp(random.random(N)**2)+1))

# account for the normalizing factor
print(2*s/N)
```

```
0.8388988085045441
```

3 Problem 2: A random point on Earth's surface

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

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

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

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

3.1 Part (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.

The ranges are $[0, \pi]$ and $[0, 2\pi)$.

The normalization for $p(\theta)$ is as follows:

$\int_0^{\pi} p(\theta)d\theta = \int_0^{\pi} \frac{sin(\theta)d\theta}{2} = 1$

The normalization for $p(\phi)$ is as follows:

$\int_0^{\pi} p(\phi)d\phi = \int_0^{2\pi} \frac{d\phi}{2\pi} = 1$

3.2 Part (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.

Choosing a random value of ϕ is simple: we generate a random number between zero and one with `random()` and then multiply it by 2π, resulting in $\boxed{\phi = 2\pi z}$ where z is the random number. Choosing a random θ is a bit harder. We want to choose from the distribution $p(\theta) = \frac{1}{2}\sin(\theta)$. By integrating over $p(\theta)$ from 0 to θ, we obtain:

$z = \frac{1}{2}\int_0^{\theta} \sin(\theta')d\theta' = \frac{1}{2}(1 - \cos(\theta))$,

where z is a uniform random number between zero and one. Rearranging for θ, this gives us $\boxed{\theta = \cos^{-1}(1 - 2z)}$. Once we have the values of θ and ϕ, we can get the Euclidean coordinates of the point from the standard transformation:

$x = r\sin(\theta)\cos(\phi), y = r\sin(\theta)\sin(\phi), z = r\cos(\theta)$

where r is the radius of the sphere.

3.3 Part (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.

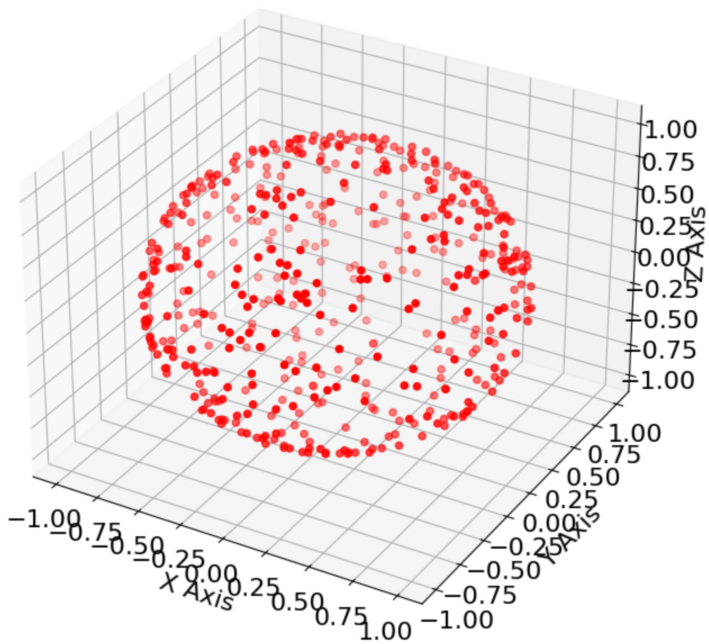
[6]:
```python
# write routine that generates random theta and phi coordinates
def random_theta_phi(N):

    # calculate theta
    theta=np.arccos(1-2*random.random(N))

    # calculate phi
    phi=2*pi*random.random(N)

    return theta, phi
```

3.4 Part (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.

[9]:

```
# define number of random points
N=500

# define sphere radius
r=1

# loop over 500 random points
theta, phi = random_theta_phi(N)

# calculate x, y, z positions
x=r*np.sin(theta)*np.cos(phi)
y=r*np.sin(theta)*np.sin(phi)
z=r*np.cos(theta)

# Creating a 3D scatter plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, c='r', marker='o')

ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')
ax.set_zlabel('Z Axis')

plt.show()
```


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Z Axis
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Y Axis
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
X Axis

```
[ ]:
```