Homework 4 Dholakia

Ravi Dholakia

October 1, 2019

1 Introduction

1.1 Overview of Monte Carlo Integration:

Monte Carlo Integration is a numerical process which is used in large computational problems which would otherwise take too long to do the "straightforward" way. This method of integration, at least in the two dimensional sense, ideally starts by plotting a large number (the larger the better, for accuracy) of points on a grid. Then, given a function, or set of conditions, the program determines which points satisfy that condition. For example, in simple integration, to find the area under a curve, the points which happen to fall under the curve would be included. The fraction of the points out of the whole area is then proportional to the fraction of the total area which the area occupies.

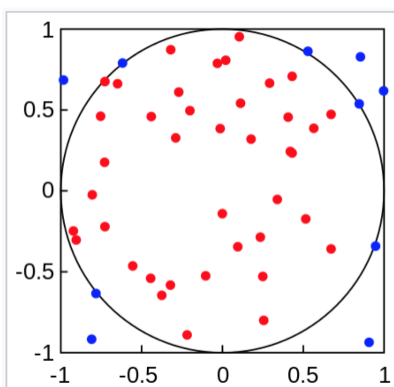


Figure 1: Red points are inside the circle, hence they satisfy the condition, whereas blue points do not.

2 Newman 10.4: Radioactive Decay Again

To integrate the area under the decay curve we can do a simple Monte Carlo calculation:

```
Integral from 0 to 5000 is:  0.948
```

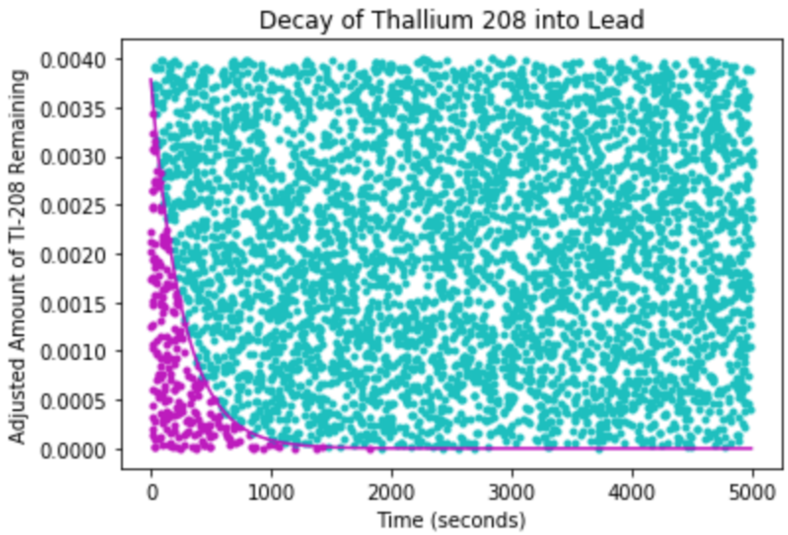


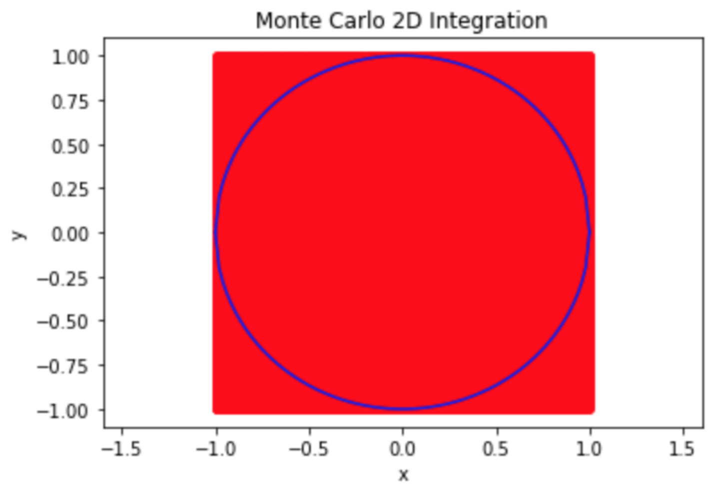
Figure 2: Points under the curve are in magenta, others are in cyan

3 Newman 10.7: Volume of a hypersphere

While a 2-D condition is simple to visualize, higher dimensions are not. However, the process is the same. For example, the Monte Carlo method for a 10-D hypersphere is similar to the method for a 2-D circle.

3.1 10-D Hypersphere: An overview of the code:

To start, I created ten dimensions, named dim1-dim10. These are arrays of length 1,000,000 (10^6), populated with zeros. Then, I created the condition which discriminates between the points included and excluded from the volume of the 10-D hypersphere. This then sets up the calculation of the volume, given that the hypersphere is unitary.




```
Area:  3.143572
```

Figure 3: The ratio of the points inside the circle to the total points is the basis of this method of integration.

3.2 10 Dimensions:

Now, we can expand in ten dimensions:

```
for i in range(N):
    if(dim1[i]**2+
       dim2[i]**2+
       dim3[i]**2+
       dim4[i]**2+
       dim5[i]**2+
       dim6[i]**2+
       dim7[i]**2+
       dim8[i]**2+
       dim9[i]**2+
       dim10[i]**2<=1):
        Hyper_Sum_Part=Hyper_Sum_Part+1
    else:
        Hyper_Sum_Part=Hyper_Sum_Part+0

print("Hypersphere volume:",1024*Hyper_Sum_Part/N)
```

Figure 4: Code to calculate volume

So the hypersphere volume comes out to:

```
Hypersphere volume: 2.557952
Percent deviation:  -0.30554945572041753 %
```

Figure 5: The volume of a unitary hypersphere

This is consistent with the formula of an n-dimensional ball:
$V_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} R^n$
So the expected volume of a unitary 10-D hypersphere is approximately: 2.55016

4 Newman 5.2 Simpson's rule

4.1

I fooled around too much with trying to program a fit-maker for the quadratic when I should have focused on the actual material. Anyways, I wrote code which first did the trapezoidal method. This yielded 4.50656. The next chunk of code dealt with Simpson's rule. From the textbook, it states:

$$
\begin{aligned}
I(a,b) &\simeq \tfrac{1}{3}h\left[f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \ldots + f(b)\right] \\
&= \tfrac{1}{3}h\left[f(a) + f(b) + 4\sum_{\substack{k \text{ odd} \\ 1\ldots N-1}} f(a+kh) + 2\sum_{\substack{k \text{ even} \\ 2\ldots N-2}} f(a+kh)\right]. \qquad (5.9)
\end{aligned}
$$

Figure 6: Formula for the Simpson method

4.2

The accuracy of the Simpson method was much higher than the trapezoid method. This is because a quadratic curve more closely resembles the natural polynomial curve of the graph than a line.

4.3

I then copied and pasted the loop to repeat the same calculation for N=100 and N=1000. Unsurprisingly, the accuracy increased every time the number of steps increased.