

PHYS 305 Assignment 5: Determining the Volume of N-dimensional Spheres via Monte Carlo Integration.

Zachary Martin¹

¹*University of Hawaii at Manoa*
(Dated: 20 February 2019)

In this assignment, we introduce the use of Monte Carlo integration to examine the volume of n-dimensional hyperspheres using a program in C++. We find the odd fact that the volume of the hypersphere reaches a peak value at dimension 5 and then decreases towards 0 thereafter. We also compare the fractional error of the calculated volumes to the dimension and find that there is a fairly constant value until dimension 8, but don't have any ideas as to why this is. It is evident that a very large number of trials must be used to get decent precision and that the random nature of this method can cause fluctuations in the error when attempting to achieve high precision.

INTRODUCTION AND OVERVIEW

While we live spatially in three dimensions, there are times where we want to integrate over more than three dimensions to solve things about a physical system, such as 6-dimensional phase space integrals over space and momentum [3]. It is important then that we learn methods to integrate over these higher dimensions. While some integrals can be straight forward and have analytic solutions, this is not always the case. One way to get around this is to use Monte Carlo methods.

The main idea of Monte Carlo integration is that we can take advantage of volume ratios of two objects, where one object contains the other (integration is equivalent to an n-dimensional volume of some function). We choose the container such that we can easily obtain the volume, such as a (hyper)cube. Then we select random points in that cube and determine whether the point is within the boundary of interest, ie. within the volume of the contained object. If we have an infinite amount of points that are truly random, then the ratio of the number of points inside the contained volume to the total number of points should be equal to ratio of the volumes. This is discussed further in the Relevant Equations section.

To demonstrate this, we begin with a simple, easy to visualize example.

Simple Case Dim = 2, 3

We learned how to implement randomness into a simulation using the `random()` function in assignment 4 [2]. Thus we can generate random points in a square of side length 2 (containing a circle of unit radius) or a cube of side length 2 (containing a sphere of unit radius) and determine the ratio of

points within a radius to the total. A simulation of this Monte Carlo integration can be visualized in Figures 1 and 2.

Clearly, a different number of points or trials will result in different ratios, and hence different calculated areas and volumes. We can see how the true volume compares to the calculated volume for different number of trials in Figure 3. We can see clear fluctuations in the fractional error at low and even intermediate values of trials. This is due to the fact that it is probable that the ratio could be close to one, given the vast majority of points happen to be within a unit radius. And so we would get a ratio that implies the volume of the sphere is approximately equal to the volume of the cube. We can see it takes until at least 10000 trials to get a fairly stable value.

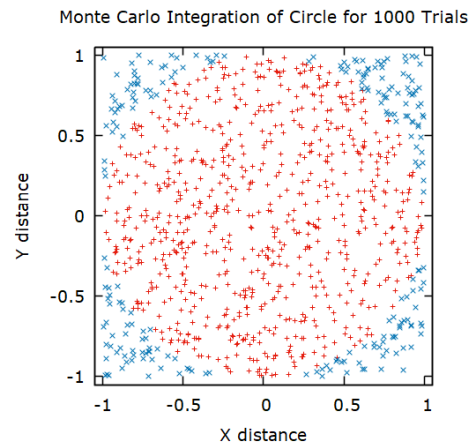


FIG. 1. 2D Case: A simulation of random points chosen within the 2x2 square. The red marks indicate a point within unit radius.

Familiar with the technique, we now generalize this to n dimensions.

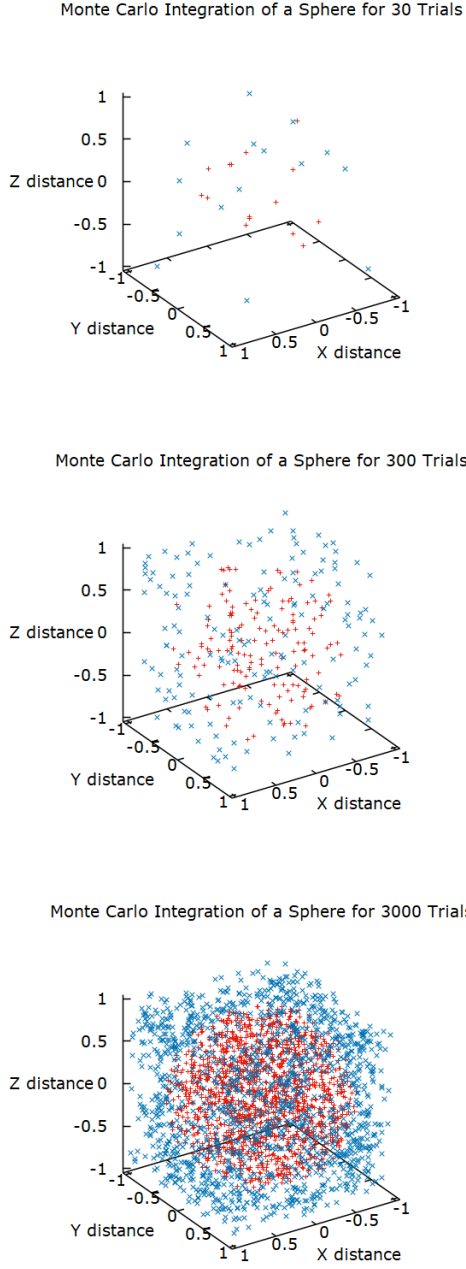


FIG. 2. 3D Case: Progression of Monte Carlo simulation for 30, 300, and 3000 trials. Like with the 2D case, the red marks indicate a point within unit radius.

RELEVANT EQUATIONS

Monte Carlo integration takes advantage of the ratio, α , of volumes of two objects where one object contains the other. Knowing the volume of the

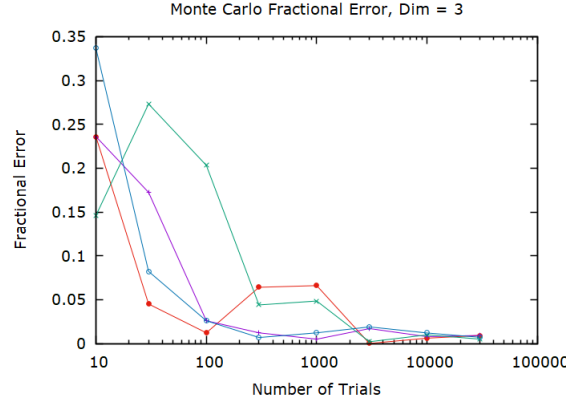


FIG. 3. Fractional error of the integrated volume of the unit sphere for varying numbers of trials. Each curve represents a different random seed used for generating the points inside the cube.

container, V_{tot} , we can get [3]

$$V = \alpha V_{\text{tot}} . \quad (1)$$

This method finds the ratio by choosing random points within the total volume and so we can get

$$\alpha = \frac{N_{\text{hit}}}{N_{\text{tot}}} , \quad (2)$$

where N_{hit} is the number of points within the volume of interest and N_{tot} is the total number of random points within the container. For calculating the volumes of hyperspheres, we contain the spheres within hypercubes, which have volume

$$V_{\text{cube}}^{(n)} = l^n , \quad (3)$$

where l is the length of a side of the cube and n is the dimension. To get a unit hypersphere ($R = 1$), we must have that $l = 2$. Thus from Equation 1, we have

$$V_{\text{sphere}}^{(n)} = \alpha 2^n . \quad (4)$$

Using a program for the Monte Carlo integration, we can find α and obtain the volume of the hypersphere. We would then like to compare this to the actual volume of the n -dimensional hypersphere $V_{\text{actual}}^{(n)}$ since we can actually analytically calculate it. To do so, we have the relation [4]

$$V^{(n)}(R) = 2 \int_0^{\pi/2} V^{(n-1)}(R \cos \theta) R \cos \theta d\theta . \quad (5)$$

We set $R = 1$ for the unit sphere and since we know the 2-dimensional volume of a sphere (area of a circle), we can generate the volumes of higher dimensions. From this recursive formula, we can derive the equation

$$V^{(n)} = \pi 2^{n-2} \prod_{i=3}^n \int_0^{\pi/2} \cos^i \theta d\theta, \quad (6)$$

which can be obtained through iterated multiplication of integrals. These integrals can be calculated using methods from previous lab assignments. The method of integration used here was the trapezoidal method from assignment 3 with 500000 intervals [1].

DESCRIPTION OF COMPUTATIONAL PROBLEM

Computational methods are the only way to tackle this problem using Monte Carlo integration. We require randomness and we need to be able to run thousands and thousands of trials.

Implementing the equations and concepts above into a C++ program is fairly straightforward. We only care about points within unit radius so we check if the distance, that is, the n-dimensional hypotenuse, is less than 1. If yes, then we count it as a hit. From there we can use Equations 2 and 4.

```
while(n<Ntrials){
    Rsq = 0.0;
    for (i=0; i<NDIM; i++){
        xi = (drand48()-0.5)*2.;
        Rsq += xi*xi;
    }
    hit += Rsq <= 1.0 ? 1.0 : 0.0;
    n++;
}
Vtot = D;
for (i=1; i<NDIM; i++){
    Vtot *= D;
}
Vsphere = hit/Ntrials*Vtot;
```

Then for the actual volume, we use the `int_trap()` function we defined in assignment 3 for the new integrand in Equation 6 [1].

```
double integproduct = 1.0;
for (int j=3; j<=NDIM; j++){
    integproduct *= int_trap(vmin, vmax, j);
    // vmin=0, vmax=M.PI/2, j is iterative index
}
Vtrue = M.PI*pow(2., NDIM-2.)*integproduct;

double f (double x, int y){
    return pow(cos(x), y);
    // y = iterative index from 3 to n
}
```

RESULTS AND GRAPHS

We start by testing the program for the simple cases, 2D and 3D. From Figure 4, we can see that the 3D curve matches the behavior from Figure 3. Again, around 10000 trials gives a fairly stable value. To be more accurate, we choose to use 500000 trials which seems to be a good choice from the figure. Then using the program for dimensions up to 10D, we obtain the values in Table I.

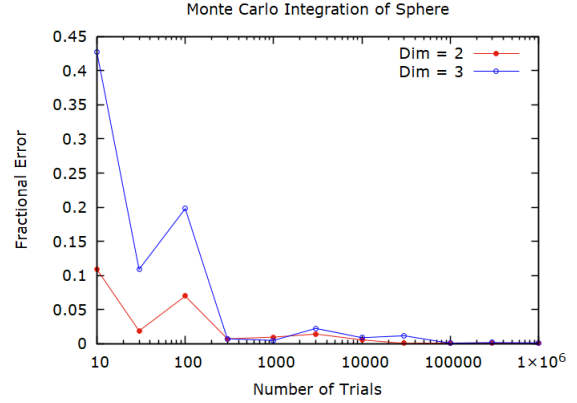


FIG. 4. Fractional error of the Monte Carlo integration of the 2D and 3D cases using the generalized program. While fluctuations are present early on, the error does decrease closer to 0 after 10000 trials. For this reason, we use 50000 trials for the determination of the values in Table I.

Dim	Vcube	Vsphere	Vactual	FracErr
2	4	3.14	3.142	0.0004484
3	8	4.192	4.189	0.0008465
4	16	4.946	4.935	0.00237
5	32	5.247	5.264	0.003194
6	64	5.182	5.168	0.002731
7	128	4.736	4.725	0.002432
8	256	4.133	4.059	0.0184
9	512	3.326	3.299	0.00832
10	1024	2.515	2.55	0.01381

TABLE I. Monte Carlo integration values of the unit hypersphere for dimensions 2 to 10. The number of trials for these values was 500000.

DISCUSSION AND ANALYSIS

We are now able to analyze the behavior of the unit hypersphere through varying dimensions. It is

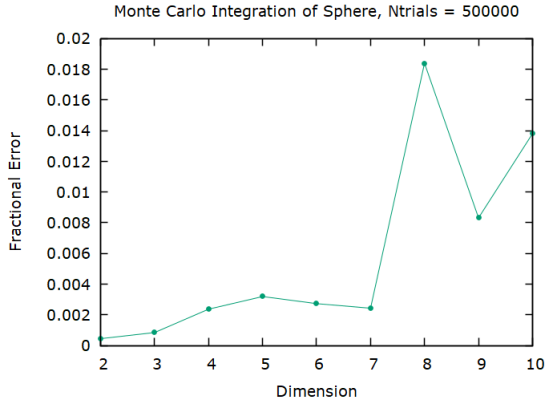


FIG. 5. The fractional error in Monte Carlo integration with varying dimension.

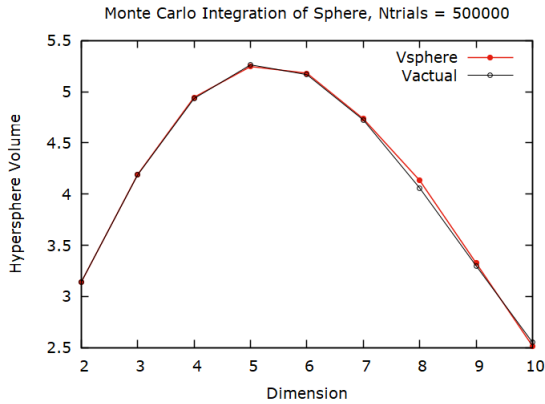


FIG. 6. Volume of the unit hypersphere in different dimensions. We can see clearly an odd fact that the volume has a peak at dimension 5, then decreases thereafter.

evident from Figure 6 that something strange happens when we reach dimension 5. The volume of the sphere actually begins to decrease, despite the volume of the cube containing it increasing. This can be inferred from Equation 6 as well. As the degree of the cosine increases, the waves become sharper but still at amplitude 1. Thus the area under the curve for high powers of cosine are less than 1 and continually getting closer to 0. So at higher dimensions, the volume has factors that are decreasing it rather than increasing. We can also see from Figure 5 that

the error is generally constant up until dimension 8. At that point, there is a significant jump in error. The height of the jump may be misrepresented here since the random nature of the calculation can cause fluctuations at such precise numbers. But there is a general increase by a factor of 10. So the relationship between the error and dimension isn't clear. I am not quite sure what causes this.

CONCLUSIONS

We were able to use Monte Carlo methods to find and analyze the volume of the unit hypersphere at dimensions higher than three. We found that the volume does not monotonically increase, but instead reaches a peak at dimension 5, then decreases towards 0 thereafter. Monte Carlo integration will be an important technique for other high dimension integration we may need to do in future problems. However, it seems we need an incredibly large number of trials to get an accurate value to fairly high precision. The random nature which causes such fluctuations in the error can be quite sensitive when looking at high precision. This implies that we must take caution in our values if we use Monte Carlo integration in that we can only expect a certain level of precision for very high numbers of trials. Nevertheless, it adds another tool in our repertoire for approximation in computational methods to solving physics problems.

ACKNOWLEDGMENTS

-
- [1] Gorham, P., "Phys305 lab 3: Numerical integration of acoustic energy," <https://laulima.hawaii.edu/x/5AbGfV> (2019), accessed on 2019-01-31.
 - [2] Gorham, P., "Phys305 lab 4: Random walks," <https://www.phys.hawaii.edu/~gorham/p305/RandomWalks.html> (2019), accessed on 2019-02-13.
 - [3] Gorham, P., "Phys305 lab 5: Monte carlo integration," <https://laulima.hawaii.edu/x/5AbGfV> (2019), accessed on 2019-02-20.
 - [4] Watkins, T., "Area-volume formulas for n-dimensional spheres and balls," <http://www.sjsu.edu/faculty/watkins/ndim.htm>, san José State University. Accessed on 2019-02-20.