Monte Carlo Integration
Lab 5

Carly Hall

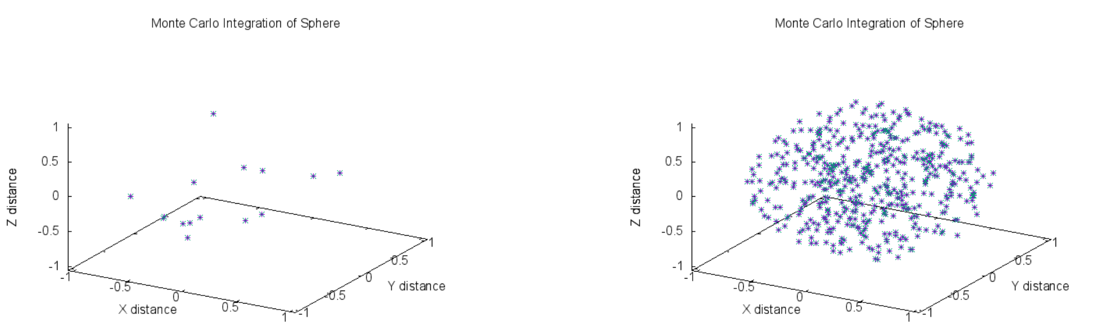
February 17, 2015

Abstract

In this lab we wrote a program to do a Monte Carlo Integration over a desired number of dimensions to find the volume of a hypersphere. We then compared this volume to the analytic integration of the hypersphere to check the validity of the program by looking at the fractional error of each dimension at different number of trials.

1 Introduction

To visualize what Monte Carlo integration is doing, imagine a pond within a square box of known area. If we randomly throw N (Ntrials) many stones in the area and count how many fall in the pond and divide it by N we will have a good estimate of the area of the pond [1]. And example of this done over a 3 dimensional sphere in Figures 1, 2, and 3 bellow.



(a) Basic integration of sphere, 30 trials

(b) Basic integration of sphere, 1000 trials

Figure 1: Monte Carlo integration of same sphere with different number of trials

With only 30 trials, less than half fell in the area giving us a very vague idea of the shape. Now with 1000 trials we have a good idea of the shape but not a clear and defined limit.

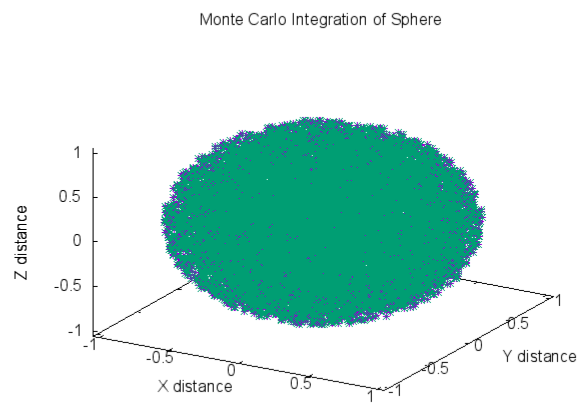


Figure 2: Basic integration of sphere, 30000 trials

with 30,000 trials we are able to see the sold sphere. Looking around the edges of the sphere, there are bumps and imperfections in the shape. We would like to see through integrations over higher dimensional volumes if this assumption holds; as the number of trials increases, the accuracy of the shape of the sphere increases.

2 Theory

Monte Carlo integration of hypersphere:

$$V_{hypercube} = (\frac{hits}{Ntrials})D^N \tag{1}$$

Analytic integration of hypersphere:

$$V_{true} = \frac{\pi^{N/2}}{(N/2+1)!} \tag{2}$$

3 Computational Apporach

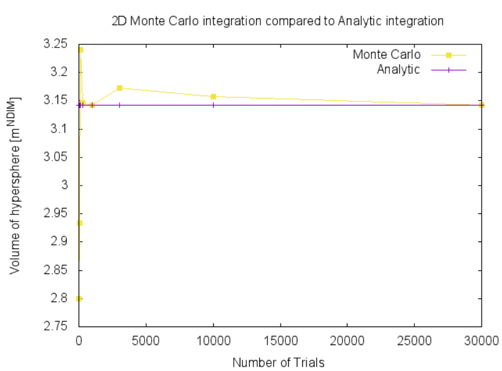
```
// P. Gorham, updated 2/10/2015 for Physics 305
// requires student completion

using namespace std;
#include <iostream>
#include <iomanip>   // required to use "setprecision" if needed
```

```
#include <fstream>
#define _USE_MATH_DEFINES
#include <cmath>

int main(int argc, char *argv[])
{

double hit,xi,R,Rsq,D,R_D,Vtot,Vsphere,Vtrue,Ferror;
int n,i,NDIM=3, Ntrials=1000000;

srand48(1299811); // a large prime

/* usage: NDIM = # of dimensions, NMAX=number of sample points */
if(argc<2){
cerr<< "usage: hypersphereMC [NDIM][NMAX]"<<endl;
exit(0);
}

NDIM = atoi(argv[1]); // number of dimensions
Ntrials = atoi(argv[2]); // number of trials



D= 2.0; // side of hypercube needed to contain hypersphere
R_D = 1.0; // radius of hypersphere
hit = 0.0; // the counter for events within sphere
n=0; // initialize the loop counter

//-------this is the main loop----------------------

while(n<Ntrials){  // continue generating coordinates up to Ntrials
    Rsq = 0.0;  // this variable accumulates the square of each coordinate
    for(i=0;i<NDIM;i++){
 xi = (drand48()-0.5)*2.; // uniform random value +/-[0,1] in each coordinate
 Rsq += pow(xi,2.);  // sum up the squares to get distance from origin
} // end of NDIM loop------------

     R = sqrt(Rsq); //  check if distance Rsq falls within R_D boundary,
     hit += R<=R_D ? 1.0 : 0.0;  // conditional expression,  increment hit counter if it does

    n++; // counter for Ntrials while loop
    }  //---------END OF WHILE LOOP-----------------

Vtot = pow(D,NDIM); // Vtot=D for 1-dimensional "hypercube" (a line)
    }
```

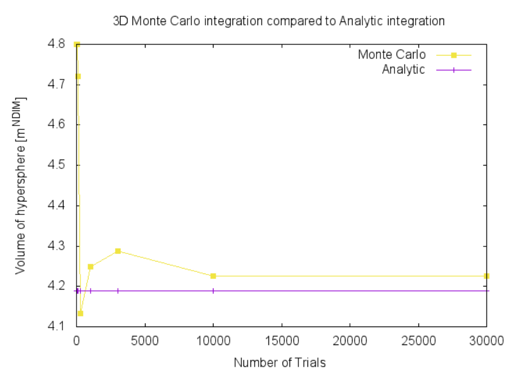
```
 /* get ratio of volumes */
 Vsphere = (hit/Ntrials)*Vtot;  // determine the hit ratio & estimated Volume

 Vtrue = pow(M_PI,NDIM/2.)/tgamma((NDIM/2.)+1.); true volume from analytic formula

 Ferror = abs(Vtrue-Vsphere)/Vtrue;

 cout << NDIM <<" dimensions, Vsphere= "<< Vsphere <<" Vtrue= "<< Vtrue <<
  "  Vtot= "<<Vtot<<" fractional error= "<< Ferror <<endl;
}
```

4 Results and Analysis

To test the accuracy of the volume produced by the Monte Carlo integration also calculated the analytic integration with every dimension and number of trials. Figure 3 through Figure 5 shows this comparison through all 30000 Ntrials.

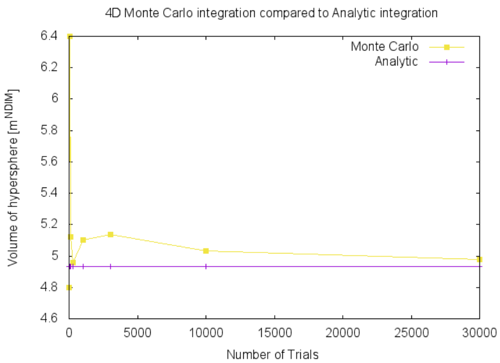


(a) 2D Monte Carlo integration compared to the analytic integral volume of a hypersphere

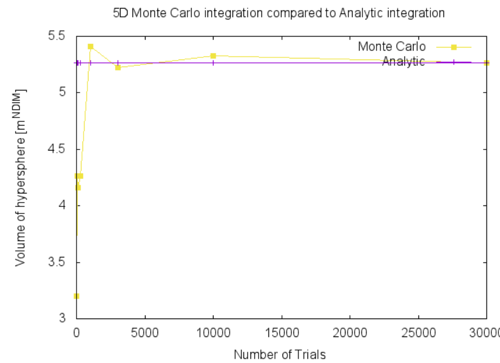


(b) 3D Monte Carlo integration compared to the analytic integral volume of a hypersphere

Figure 3: Monte Carlo integration compared to analytic integration of hypersphere with different dimensions

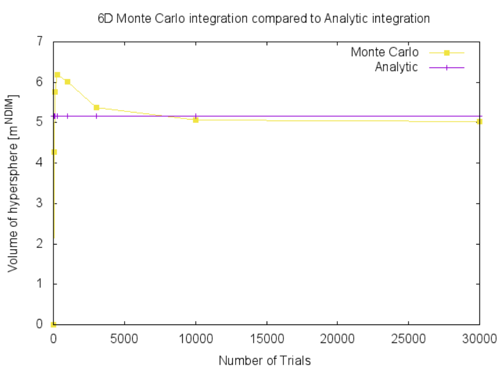


(a) 4D Monte Carlo integration compared to the analytic integral volume of a hypersphere

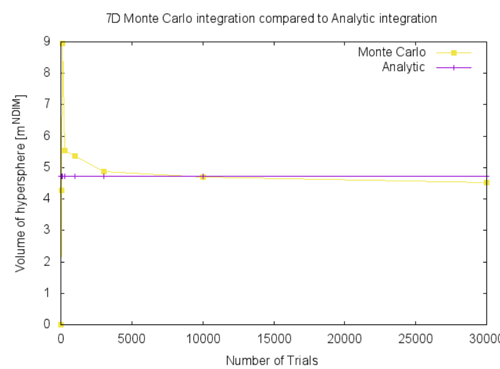


(b) 5D Monte Carlo integration compared to the analytic integral volume of a hypersphere

Figure 4: Monte Carlo integration compared to analytic integration of hypersphere with different dimensions



(a) 6D Monte Carlo integration compared to the analytic integral volume of a hypersphere



(b) 7D Monte Carlo integration compared to the analytic integral volume of a hypersphere

Figure 5: Monte Carlo integration compared to analytic integration of hypersphere with different dimensions

Volume and Fractional Error	2D	3D	4D
Monte Carlo	3.1568 ± 0.000002338	4.22507 ± 0.00866037	4.97973 ± 0.00910495
Analytic	3.14159	4.18879	4.9348
Volume and Fractional Error	5D	6D	7D
Monte Carlo	5.26613 ± 0.000445367	5.0304 ± 0.0265713	4.5184 ± 0.0436775
Analytic	5.26379	5.16771	4.72477

Table 1: Volume and Fractional error of Monte Carlo and analytic integration after 30,000 trials

Figure 6 through figure 9 is of the fractional error with increasing number of trials.

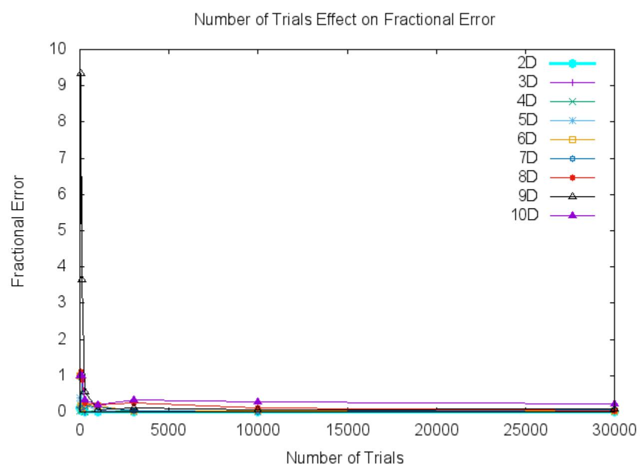


Figure 6: Fractional error of volume compared to number of trials

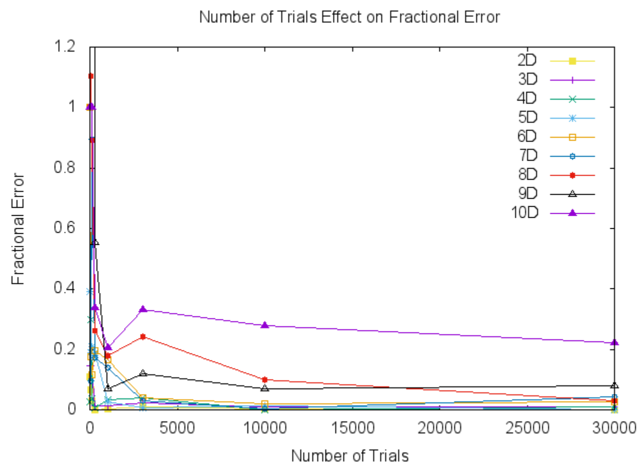


Figure 7: Fractional error of volume compared to number of trials zoomed in to look at initial divergence

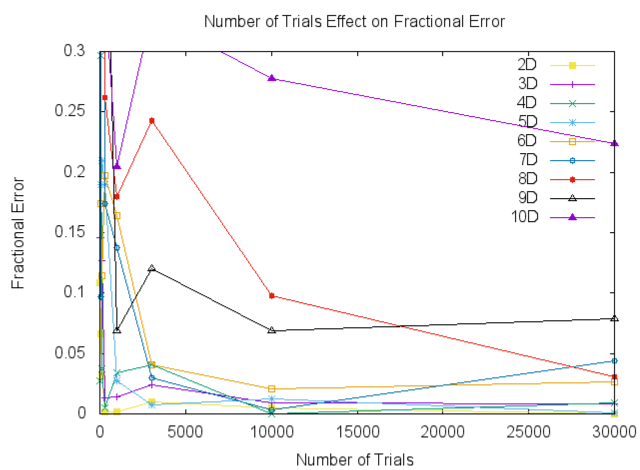


Figure 8: Fractional error of volume compared to number of trials zoomed in to look at final divergence

5 Conclusions

The resulting comparison of the Monte Carlo and analytic volume of a hypersphere proved that the Monte Carlo held true for only the 2 dimensional circle. Although for dimensions 4th, 5th and 6th dimensions. To get the precision in the order of 10^{-5} for a 10 dimensional hypersphere Ntrials would have to be set to 100,000,000.

Acknowledgement

Corey Mutnik - coding help

References

[1] $< http://www.phys.hawaii.edu/\ gorham/P305/MonteCarlo1.html >$