PHY407 Final Project: Exploring the MISER Algorithm

Mitchell Barrett

18 December 2020

1 Introduction

In this report, we do not explicitly solve a physical problem using a computational technique. We instead outline a computational technique for computing arbitrary multidimensional integrals. This, of course, relates to many different areas of physics, as numerically evaluating integrals is one of the most common problems in physics. The code that we build, then, serves as a mathematical basis for handling physical phenomena with computers.

The purpose of this project is to recreate and explore a method of Monte Carlo integration called recursive stratified sampling, with the intention of comparing it to more standard Monte Carlo methods. Recursive stratified sampling improves upon both standard Monte Carlo integration and the importance sampling method: it allows you to randomly choose points near regions of high variance in the integrand without specifying a probability distribution, making it more accurate than standard Monte Carlo integration and more versatile than importance sampling.

The particular algorithm on which we focus is called the MISER[1] algorithm (Press & Farrar, 1990), which is perhaps the most common and most well-known method of recursive stratified sampling. We will use this algorithm to calculate a few simple integrals, in both one dimension and multiple dimensions—the algorithm itself is generalized, suitable for computing any multidimensional integral. We will compare this particular algorithm to the simpler Monte Carlo methods explained in the course textbook (Newman, 2012).

2 Methods

In this section, we detail the aforementioned Monte Carlo methods. Since the more basic Monte Carlo methods were covered previously in PHY407, our discussion of these methods will be rather brief, intended to be a quick refresher rather than a full explanation of the methods. The explanations provided in §2.1–§2.3 are informed by Newman (2012). Following this, the entirety of §2.4 is informed by Press & Farrar (1990).

2.1 Monte Carlo Integration

Monte Carlo integration is a method—or, more accurately, a set of methods—for calculating integrals without using more involved integration techniques like the Simpson's rule of the trapezoidal rule. The benefit of this is that this allows one to approximate integrals that would theoretically require an infinite number of slices of infinitesimal width for the aforementioned rules to work properly.

The most basic method of Monte Carlo integration involves, simply, randomly sampling N points in a volume and calculating the fraction of those points which fall below a function f. Given enough random samples, the value of the integral can then be estimated as the fraction of points that fell below the curve multiplied by the total volume of the space from which the points were sampled. This method is not particularly good or efficient, and requires a large number of sample points to be accurate. These problems are fixed by the methods outlined in §2.2 and §2.3. In addition to these problems, this basic method does not scale into higher dimensions very well, a point of particular importance for the MISER algorithm.

[1]It's not clear why the algorithm is called MISER. I choose to believe it stands for **M**onte Carlo **I**ntegration via **S**tratified Sampling, with Nearly **E**ndless **R**ecursion.

2.2 The Mean Value Method

The mean value method is perhaps the simplest (and, according to Newman (2012), the most common) way of improving the basic Monte Carlo integration. In order to evaluate the 1D integral

$$I = \int_a^b f(x)dx \tag{1}$$

we recognize that we can write

$$\langle f \rangle = \frac{I}{b-a} \tag{2}$$

which can be rewritten to solve for I:

$$I = (b-a)\langle f \rangle \simeq \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \tag{3}$$

In the above equation, x_i represents a point sampled uniformly at random from within the bounds of integration. This can be generalized to higher dimensions with ease:

$$I \simeq \frac{V}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \tag{4}$$

The error of this approximation is given by

$$\sigma = V\sqrt{\frac{\mathrm{var} f}{N}} \tag{5}$$

While simple, this method still requires a lot of sample points to be accurate, and does not generalize to higher dimensions efficiently.

2.3 Importance Sampling

Newman (2012) states that one problem with basic Monte Carlo integration is that it does not work particularly well for functions containing a divergence. The workaround for this issue is to sample points non-uniformly, specifying some probability distribution from which to draw points. For a function $g(x)$, the *weighted* average between a and b is defined:

$$\langle g \rangle_w \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} \tag{6}$$

where $w(x)$ is our weight, which can generally be anything (though in practice, not everything is useful). To calculate the integral of $f(x)$, we set $g(x) = f(x)/w(x)$ to obtain

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x)dx \tag{7}$$

From this equation, the explicit formula for the Monte Carlo approximation is derived to be:

$$I \simeq \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x)dx \tag{8}$$

where the N points are sampled randomly from the probability distribution $p(x)$, given by

$$p(x) = \frac{w(x)}{\int_a^b w(x)dx}. \tag{9}$$

The error associated with this method is given by

$$\sigma = \sqrt{\frac{\mathrm{var}_w(f/w)}{N}} \int_a^b w(x)dx \tag{10}$$

This provides some advantages over the mean value method, reducing the error of integrals whose integrands contain divergences or high variances. However, this raises the annoying problem of choosing a function $w(x)$. If $w(x)$ is not a "simple" function, then one must also do work to determine $p(x)$ and generate a transformation between uniformly-sampled points and points sampled with probability $p(x)$. These problems make this method somewhat difficult to apply. In particular, it is hard to generalize this method to higher dimensions—whilst the equations can be written out with ease, it makes choosing a useful $w(x)$ harder.

2.4 The MISER Algorithm

The MISER algorithm represents a Monte Carlo method for computing integrals that works effectively for any dimension and any given function. One does *not* need to choose a probability distribution from which to sample points, providing an improvement over importance sampling. In addition, due to how the randomly-sampled points are chosen, the error is lower than that of the basic mean value method.

This method implemented by the MISER algorithm is called *recursive stratified sampling*. "Stratified sampling" means the volume of integration is partitioned, and the randomly-sampled points are chosen according to which regions have the highest variance. The recursion comes into play when stratifying the volume of integration; the new volumes created by the partitions are themselves partition further. This partitioning is done until each volume contains a pre-selected number of points. This method therefore divides N total points up across the volume of integration according to where the integrand varies the most. This phenomenon will be demonstrated in §3. The following subsections will explain the mathematics governing the algorithm described, as well as a discussion of error estimates and a brief handling of runtime. All following explanations reference Press & Farrar (1990).

2.4.1 Evaluating the Integral

The mathematical basis for the MISER algorithm begins quite simply: the simplest way to estimate the mean of a function given N inputs sampled uniformly at random is

$$\langle \tilde{f} \rangle = \frac{1}{N} \sum_{i=1}^{N} f_i \tag{11}$$

where f_i is f evaluated at the i'th random point. An estimate for the integral of the function would be this value multiplied by the volume of integration. Currently, this is precisely the mean value Monte Carlo method. However, we seek to improve upon the mean value method by selecting our random points in a better way. Consider partitioning our volume of integration, V, into two equal subvolumes V_a and V_b. We can now estimate the quantity from Eq. (11) as

$$\langle \tilde{f} \rangle' \equiv \frac{1}{2} \left[\langle \tilde{f} \rangle_a + \langle \tilde{f} \rangle_b \right] \tag{12}$$

This quantity is the average of the averages in the subvolumes. The important insight is as follows: the number of points sampled from V_a and from V_b do *not* need to be equal. We therefore choose to divide our N total points into N_a points to be sampled from V_a and N_b points to be sampled from V_b. Press & Farrar (1990) specify that the variance of the estimator $\langle \tilde{f} \rangle'$ is minimized when

$$\frac{N_a}{N} = \frac{\sigma_a}{\sigma_a + \sigma_b} \tag{13}$$

We will discuss how to calculate the error estimates σ_a and σ_b in the following section. For now, however, this gives us a way to decide how many points to assign to each volume. This is, in essence, the end of the algorithm's mathematical step. The only remaining step is the recursion. A maximum number of points,

N_{min} is specified. This is the number of points below which a simple Monte Carlo integral will be done. While $N_a > N_{\text{min}}$, the volume V_a will be partitioned again, and N_a will be divided up into the new partitioned volumes according to Eq. (13). This turns the integral into a series of many small mean value Monte Carlo integrals. Doing many small integrals in this manner improves the error in the calculation. This method is also completely generalized, working on integrals of any dimension.

One point that was not mentioned was how to make the choice of V_a and V_b. At each recursive step, the algorithm will attempt to partition the volume in half along each coordinate axis. A fixed, given number of points are used to calculate rough Monte Carlo integrals in each subvolume. The partition which minimizes the overall variance is chosen to be the best partition at that step of recursion.

Once the recursion is done down to the lowest level, and each subvolume contains at most N_{min} points, a mean value Monte Carlo integral is done on each volume, and the results are added together and averaged. This generates the estimate for the value of the integral.

2.4.2 Estimating the Error

We start with the standard definition for the variance of a function f:

$$\text{var} f \equiv \langle f^2 \rangle - \langle f \rangle^2 \tag{14}$$

We adopt the notation

$$\text{var}\langle \tilde{f} \rangle = \frac{\text{var} f}{N} \tag{15}$$

implying

$$\text{error} = V \sqrt{\text{var}\langle \tilde{f} \rangle}. \tag{16}$$

Upon applying Eq (12) in the numerical calculation, we obtain the following expression:

$$\text{var}\langle \tilde{f} \rangle' = \frac{1}{4} \left[\text{var}\langle \tilde{f} \rangle_a + \text{var}\langle \tilde{f} \rangle_b \right] \tag{17}$$

Press & Farrar (1990) go on to prove that this expression is always smaller than the simple error for a basic Monte Carlo mean value estimate of the integral using the same number of points. Currently, though, we are only interested in the method by which this algorithm estimates its own error, so we will not go through the algebra to prove this fact.

One can see from the form of the previous equations that $\text{var}\langle \tilde{f} \rangle'$ can be calculated recursively at each step, with the final error being given by Eq (16).

2.4.3 Runtime

At each step of recursion, the algorithm tests partitioning the volume along each coordinate axis. This means that the runtime of the algorithm is proportional to the dimension d over which one integrates. A three-dimensional integral will require three times as many calculations per recursion as a one-dimensional integral. For k steps of recursion, then, the runtime goes like kd.

3 Results

As mentioned, this algorithm was not used to explicitly solve a physics problem. Instead, calculated values for a number of test cases are presented along with error estimates. For one case, a comparison is made between the MISER algorithm and the mean value method, to demonstrate the improvement in error. Additionally. as mentioned in §2.4, the N total points are divided across the volume according to where f varies the most. The following section offers a more complete explanation of this point.

3.1 Distribution of Random Samples

At each recursive step, since points are distributed according to Eq (13), more points are placed into subvolumes in which the integrand has a higher variance. Across regions of gradual change, only a small fraction of the total points will be placed. Across regions where the integrand varies dramatically, many more points are placed, giving a more accurate picture of the highly variable regions of the function. This is demonstrated through integrating the 2D unit circle to find its area. In the case of the unit circle, the integrand is:

$$f(x) = \begin{cases} 1 & x^2 + y^2 \leq 1 \\ 0 & x^2 + y^2 > 1 \end{cases} \tag{18}$$

This equation varies drastically near $x^2 + y^2 = 1$, and does not vary at all in the interior and exterior regions. Fig 1 shows the points sampled by the MISER algorithm when integrating this function to determine the area of the unit circle. This demonstrates the way in which the MISER algorithm causes randomly-sampled to cluster around regions of higher varaince.

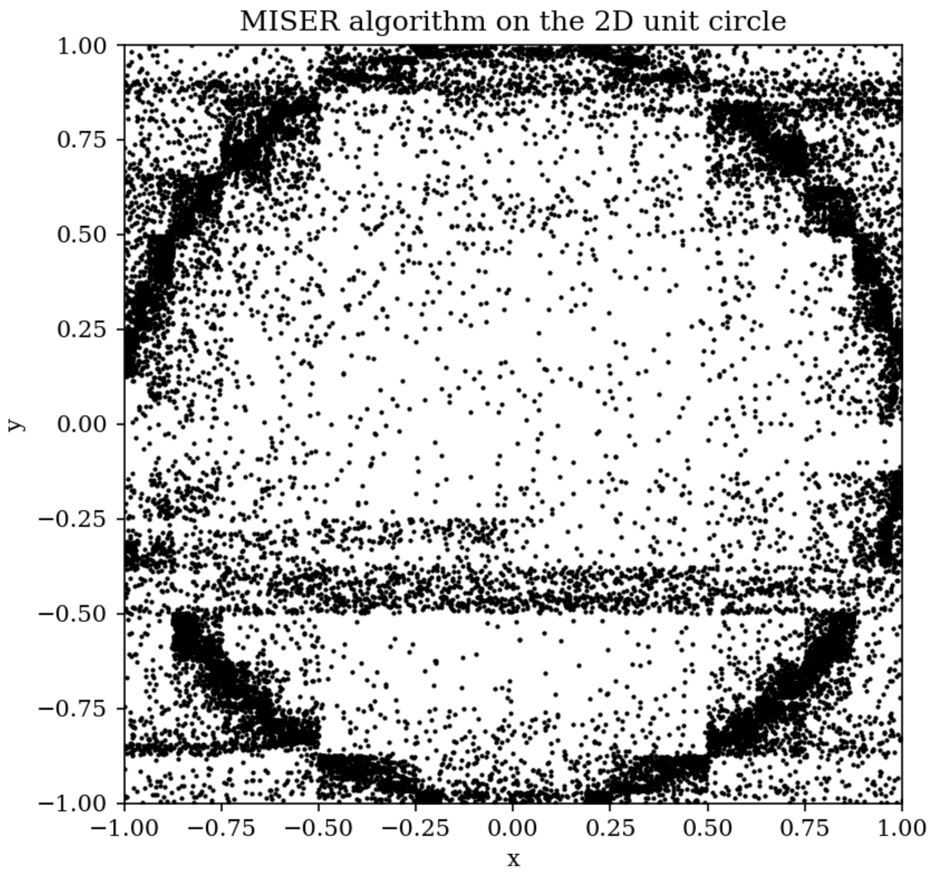


Figure 1: A plot of the 25000 points randomly sampled by the MISER algorithm in calculating the area of the unit circle. This demonstrates how the MISER algorithm causes points to be sampled from near areas of high variance in the function.

3.2 Test Cases

As test cases to ensure the correctness of the algorithm, MISER was used to calculate the area of the unit circle, the volume of the unit sphere, and the 4D volume of the 4D unit hypersphere. These volumes are known to be

$$V_{\mathrm{2D}} = \pi \approx 3.14159 \tag{19}$$

$$V_{\mathrm{3D}} = \frac{4\pi}{3} \approx 4.18879 \tag{20}$$

$$V_{\mathrm{4D}} = \frac{\pi^2}{2} \approx 4.93480 \tag{21}$$

Comparing the known values to the values computed by MISER:

$$V_{\mathrm{2D}} = 3.14463 \pm 0.00322 \tag{22}$$

$$V_{\mathrm{3D}} = 4.18903 \pm 0.00576 \tag{23}$$

$$V_{\mathrm{4D}} = 4.94234 \pm 0.01329 \tag{24}$$

we can see that the algorithm does indeed return sensible values. MISER was also tested on the integral

$$\int_{\phi=0}^{2\pi} \int_{r=0}^{1} \ln(r)(r dr d\phi) \tag{25}$$

This integral can be directly computed to find

$$\int_{0}^{2\pi} \int_{0}^{1} \ln(r) r dr d\phi = -\frac{pi}{2} \approx -1.57080 \tag{26}$$

Using MISER, the calculated value of this integral was

$$\int_{0}^{2\pi} \int_{0}^{1} \ln(r) r dr d\phi = -1.57030 \pm 0.00034 \tag{27}$$

This test case also demonstrates the correctness of the algorithm. Terminal outputs are presented in Fig 2 to verify the truthfulness of the values given in this section..

3.3 Comparing MISER to the Mean Value Method

The 1D integral of $\ln(x)$ was calculated between $x = 0$ and $x = 1$ using both the MISER algorithm and the mean value method. Both calculations were done with only 1000 randomly sampled points. The value of this integral is known to be -1, a value that we may compare to the results of the two methods. Using the mean value method, the value of the integral was found to be

$$\int_{0}^{1} \ln(x) dx = -1.018 \pm 0.031 \tag{28}$$

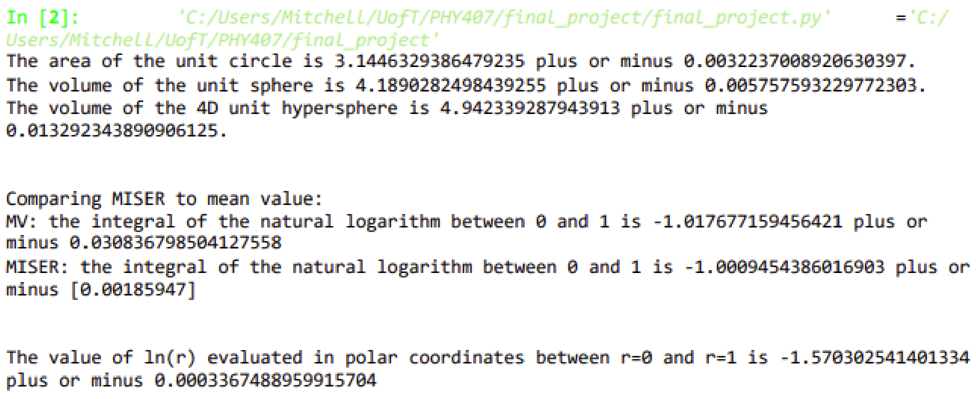
whereas with MISER, this was calculated to be

$$\int_{0}^{1} \ln(x) dx = -1.001 \pm 0.002 \tag{29}$$

demonstrating the improved accuracy of the MISER algorithm. Over several runs, the MISER value was also much less variable than the mean value estimate.

4 Conclusion

This MISER algorithm represents an efficient and easy-to-implement method of calculating multidimensional integrals. It provides significantly more accurate results than simple Monte Carlo mean value estimates do, and the recursive method by which it samples random points means it can be used to sample highly variable integrands without needing to determine a separate probability distribution, as one would with importance sampling. While we did not solve any physical probelms in this report, this algorithm may be implemented to solve any complex physical problem involving a multidimensional integral.



```
In [2]:        'C:/Users/Mitchell/UofT/PHY407/final_project/final_project.py'      ='C:/
Users/Mitchell/UofT/PHY407/final_project'
The area of the unit circle is 3.1446329386479235 plus or minus 0.0032237008920630397.
The volume of the unit sphere is 4.1890282498439255 plus or minus 0.005757593229772303.
The volume of the 4D unit hypersphere is 4.942339287943913 plus or minus
0.013292343890906125.


Comparing MISER to mean value:
MV: the integral of the natural logarithm between 0 and 1 is -1.017677159456421 plus or
minus 0.030836798504127558
MISER: the integral of the natural logarithm between 0 and 1 is -1.0009454386016903 plus or
minus [0.00185947]


The value of ln(r) evaluated in polar coordinates between r=0 and r=1 is -1.570302541401334
plus or minus 0.0003367488959915704
```

Figure 2: Outputs from one run of the code implementing the MISER algorithm.

References

Newman, M. (2012). *Computational Physics*. University of Michigan.

Press, W. H., & Farrar, G. R. (1990). Recursive stratified sampling for multidimensional monte carlo integration. *Computers in Physics*, *4*, 190–195.
URL https://aip.scitation.org/doi/abs/10.1063/1.4822899