# USING MONTE CARLO INTEGRATION AND CONTROL VARIATES TO ESTIMATE $\pi$

NICK CANNADY, PAUL FACIANE, AND DAVID MIKSA

ABSTRACT. We will demonstrate the utility of Monte Carlo integration by using this algorithm to calculate an estimate for $\pi$. In order to improve this estimate, we will also demonstrate how a family of covariate functions can be used to reduce the variance. Finally, the optimal covariate function within this family is found numerically.

## 1. AN INTRODUCTION TO MONTE CARLO INTEGRATION

1.1. **Monte Carlo Integration.** Monte Carlo Integration is a method for approximating integrals related to a family of stochastic processes referred to as Monte Carlo Simulations. This term was coined in the mid 1950's by Nicholas Metropolis, one of the original scientists of the Manhattan Project, to reference the random processes of this class of algorithms. The method relies on the construction of a random sample of points so outputs are non-unique; however, these outputs probalistically converge to the actual value of the integral as the number of sample points is increased. Since its development, Monte Carlo Integration has been used to evaluate many problems which otherwise become computationally inefficient or unsolvable by other methods.

1.2. **Algorithm.** To evaluate $I = \int_a^b f(x)\,dx$ by Monte Carlo Integration, first generate a sequence of N uniformly distributed random variables within the interval. That is, create $X_i \sim U[a,b]$ and let $Y_i = f(X_i)$ for $1 \leq i \leq N$. Find the average $\bar{Y}_N$ and multiply this value by the length of the interval, $(b-a)$, for an approximation of $I$. Of course, keep in mind that in general larger choices of N provide better approximations for $I$.

An example of this process is tossing rocks into a circular pond for an estimation of $\pi$. If we enclose a circular pond of radius $r = 1$ with a square having sides of length 2, we will see that $A_{square} * \frac{n}{N} \approx \pi$ where $n$ is the number of rocks in the pond and $N$ is the number of rocks within the square. Later, we propose another method to estimate $\pi$ with MCI.

### 1.3. Derivation.

**Definition 1.1.** [1] For any continuous random variable $X \sim \rho(X)$ and $Y = f(X)$, the **expected value** of Y is defined as:

$$E[Y] = E[f(X)] = \int_{-\infty}^{\infty} f(x)\rho(x)\, dx.$$

If we take $\rho(x)$ to be the uniform probability density function on $[a, b]$ so that

$$\rho(x) = \begin{cases} \frac{1}{b-a} & \text{when } x \in [a, b], \\ 0 & \text{otherwise} \end{cases}$$

then $E[Y]$ takes the form $\int_a^b f(x) \frac{1}{b-a}\, dx$. Hence,

$$I = \int_a^b f(x)\, \frac{b-a}{b-a}\, dx = (b-a) \int_a^b f(x)\, \frac{1}{b-a}\, dx = (b-a)E[Y].$$

**Theorem 1.2.** [2] ***The Law of Large Numbers*** *states that for any random variable $X$ with $E[X] = \mu_X$, that $\bar{X}_N \xrightarrow{P} \mu_X$ as $N \to \infty$.*

Because $I$ can be expressed in terms of $E[Y]$, this means

$$(b-a)\bar{Y}_N \xrightarrow{P} (b-a)\mu_Y = (b-a)E[Y] = I.$$

Thus, we can say for large N, $(b-a)\bar{Y}_N = (b-a)\bar{f}(X_N) \approx I$.

### 1.4. Analyzing MCI.

Many of the details of MCI should seem strikingly similar to the process of Riemann integration. In both cases, we choose an arbitrary selection of points across the particular interval in mind, and use these values to construct a sum which proves more precise as the number of points is increased. We will now look at this more precisely.

**Theorem 1.3.** ***Riemann Integrability***
*$\int_a^b f(x)\, dx$ exists and equals I if and only if*

$$\forall\, \epsilon > 0, \exists\, \delta > 0 \textit{ where} \left| \left( \sum_{i=1}^{N} f(\tilde{x}_i)\Delta x_i \right) - I \right| < \epsilon$$

$$\forall\, X = \{x_1, x_2, ..., x_{N+1}\} \textit{ with } a \leq x_1 < x_2 < ... < x_{N+1} \leq b \textit{ and}$$

$$\Delta x_i = x_{i+1} - x_i < \delta \textit{ for } 1 \leq i \leq N \textit{ and } \forall\, \tilde{X} = \{\tilde{x}_i | \tilde{x}_i \in [x_i, x_{i+1}]\}.$$

So assuming $I$ exists, $\sum_{i=1}^{N} f(\tilde{x}_i)\Delta x_i \approx I$ as long as the partition $X$ and the set of evaluation points $\tilde{X}$ are properly constructed. Assume $\tilde{X} = \{X_1, X_2, \ldots, X_N\}$ is generated from the uniform probability distribution and assume $X$ is a partition satisfying the above qualities. Since the points are to be uniformly distributed, we may assume for large N, $\Delta x_i \approx \Delta x_j \approx \frac{b-a}{N}$ for $1 \leq i < j \leq N$. Hence,

$$\sum_{i=1}^{N} f(\tilde{x}_i)\Delta x_i \approx \sum_{i=1}^{N} f(\tilde{x}_i)\Delta x_0 \approx \Delta x_0 \sum_{i=1}^{N} f(\tilde{x}_i) \approx \frac{b-a}{N}\sum_{i=1}^{N} f(\tilde{x}_i) = (b-a)f\bar{(}x).$$

Thus, MCI is simply a restructured form of the Riemann Sum. One should note that larger choices of N give better approximations, since each additional point creates a further refinement of the partition $X$.

### 1.5. **Sample Size and Its Relation to the Mesh of $X$.**

**Definition 1.4. The Mesh of $X$**
Denoted $\|X\|$, the mesh of a partition, $X$, is defined as the $max\{\Delta x_i = x_{i+1} - x_i\}$ for $1 \leq \ i < N$ where N is the size of the partition.

Since for a set function and interval, the error of the Riemann Sum, $\epsilon$, depends upon the size of the mesh, which is strictly less than $\delta$, one should pay special attention to the behavior of the mesh during Monte Carlo simulations. The sole part of the Monte Carlo algorithm pertaining to the mesh is the random generation of the sample $\{X_1, X_2, ..., X_{N'-1}\}$. Thus the question becomes: if an interval is divided into $N'$ subintervals by $N' - 1$ points chosen from the uniform distribution over that interval, what is the probability that no single subinterval is larger than $\delta$?
This is the same as requiring the length of exactly $N'$ subintervals to be less than $\delta$. For $i = 1, 2, ..., N' - 1$, it can be shown that the probability that none of $i$ specific subintervals will be less than $\delta$ is equal to $(1 - i\delta)^{N'-1}$[3]. Thus, the probability that the mesh of a partition is less than some given $\delta$ is given by

$$[3] \quad 1 - \sum_{i=1}^{r} (-1)^{i-1} \binom{N'}{i} (1 - i\delta)^{N'-1} \text{ where } r = \left\lfloor \frac{1}{\delta} \right\rfloor$$

which will show to increase in value as $N'$ is increased. Thus, larger values of $N'$ yield smaller values of $\epsilon$ which in turn sharpens the approximation. We will later show yet another method for bettering the approximation.

## 2. Use of Monte Carlo Integration to Estimate $\pi$

We can now compute an estimate for the value of the definite integral $\int_{-1}^{1} \frac{1}{1+x^2}\,dx$ using Monte Carlo Integration and use this to estimate the value of $\pi$. This is possible since it is known from calculus that

$$\int_{-1}^{1} \frac{1}{1+x^2}\,dx = \frac{\pi}{2}$$

In order to use Monte Carlo Integration, first we define $X$ to be a random variable uniformly distributed on the interval [0,1], that is $X \sim U[0,1]$. Next we let $f(X) = \frac{1}{1+X^2}$ which is a function of our random variable. By definition the expected value of $f(X)$ is

$$E[f(X)] = \int_{-\infty}^{\infty} f(x)g(x)\,dx$$

where $g(x)$ is the probability distribution of $X$. Using the definitions from Section 1 we see further that

$$E[f(X)] = \int_{-1}^{1} \frac{1}{1+x^2}\frac{1}{2}\,dx$$

or

$$E[f(X)] = \int_{0}^{1} \frac{1}{1+x^2}\,dx$$

since we are dealing with an even function. Thus, $2E[f(X)]$ is equal to the value of the desired definite integral. Now we use a simulation to estimate $E[f(X)]$. This is done by instructing Mathematica to repeatedly pick a random number between 0 and 1 to use as $X$ and then record the value of $f(X)$. Once this has been done one hundred thousand times, the mean is taken as an estimate of $E[f(X)]$. Recall that this is justified by the Law of Large Numbers explained previously. Finally, we multiply this estimate by 2 to get our estimate of the value of the desired definite integral. Doing this in Mathematica yields an estimate of 1.5713 which is fairly close to the known value of the definite integral $\frac{\pi}{2} \approx 1.5708\ldots$ Furthermore, we can double our estimate and obtain 3.14261, a fairly close estimate of $\pi$.

## 3. Variance Reduction

Variance reduction refers to a variety of methods which may be employed in conjunction with Monte Carlo simulations, including partial integration, systematic sampling, and control variates. In order to fully explain the following concepts, a few definitions must be established.

**Definition 3.1. Variance**
[1] If $X$ is a random variable with mean $\mu_X$, the variance of $X$, $Var(X)$, is defined by

$$Var(X) = E[(X - \mu_X)^2]$$

**Definition 3.2. Covariance**
[1] Let $X$ and $Y$ be random variables. The covariance between X and Y, denoted Cov(X,Y), is defined by

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

**Definition 3.3. Correlation**
[1] The correlation of two random variables $X$ and $Y$, denoted by $\rho(X, Y)$, is defined as

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

as long as $Var(X)Var(Y) > 0$. It can be shown that $-1 \leq \rho(X, Y) \leq 1$

The joint goal of the aforementioned variance reduction methods is to minimize the variance on a simulation. The variance in a simulation represents the statistical uncertainty in the result. Thus, reduction of variance clearly leads to a more accurate result. In this paper, we are interested in demonstrating a method using what are known as control variates and testing the efficacy of the control variates method.

3.1. **Control Variates.** The control variate method is useful when trying to simulate the expected value of a random variable, X. A second random variable, Y, for which the expected value is known, is introduced. The correlation between the two random variables must then be maximized such that the variance of the estimate of the X is reduced, leading to a more accurate simulation.

3.2. **Derivation of Formulae.** Suppose $X$ is a random variable and that we wish to simulate $E[f(X)]$. Suppose also $\exists$ $g(X)$ such that $E[g(X)] = \mu_g$. We then define a variable

$$[2] \quad W = f(X) + a[g(X) - \mu_g]$$

Note that

$$[2] \quad E[W] = E[f(X) + a[g(X) - \mu_g]] = E[f(X)]$$

Note that the variance of $W$ is

$$[2] \quad Var(W) = Var[f(X)] + a^2 Var[g(X)] + 2aCov[g(X), f(X)]$$

The optimal value of $a$ can be found using simple calculus by first differentiating with respect to $a$,

$$\frac{d}{da}[Var(W)] = \frac{d}{da}[Var[f(X)] + a^2 Var[g(X)] + 2aCov[g(X), f(X)]]$$

setting the derivative to 0,

$$0 = 2aVar[g(X)] + 2Cov[g(X), f(X)]$$

and solving for $a$,

$$a = -\frac{Cov[g(X), f(X)]}{Var[g(X)]}$$

We substitute this value of $a$ into our formula for $Var(W)$ and get

$$Var(W) = Var[f(X)] - \frac{[Cov[g(X), f(X)]]^2}{Var[g(X)]}$$

We further define

$$R(\sigma) = \frac{[Cov[g(X), f(X)]]^2}{Var[g(X)]}$$

for notation convenience.

3.3. **Family of $g_\sigma(X)$.** In the variance reduction of our simulation, we used the family of functions

$$g_\sigma(X) = e^{\frac{-X^2}{\sigma}}$$

for

$$\sigma > 0$$

The parameter $\sigma$ must be optimized to determine which $g_\sigma(X)$ would most reduce the variance of our estimate.

3.4. **Optimization of $\sigma$.** We saw in the previous sections that

$$Var(W) = Var[f(X)] - \frac{[Cov[g(X), f(X)]]^2}{Var[g(X)]}$$

or

$$Var(W) = Var[f(X)] - R(\sigma)$$

We have no control over the value of $Var[f(X)]$ itself, due to its constancy. However, we if we can maximize the value of $R(\sigma)$, then we would minimize $Var(W)$. In order to analytically optimize $R(\sigma)$, we need to differentiate the term. However, we found the term intractable and optimized the term statistically.

## 4. Numerical Optimization of Variance

We have shown to most efficiently reduce the variance of the approximation, it suffices to maximize $R(\sigma) = \frac{Cov(f(x),g(x,\sigma))^2}{Var(g(x,\sigma))}$. To numerically calculate the optimal $\sigma$, rewrite the ratio in mind and use numerical methods to plot its value for a range of values.

$$R(\sigma) = \frac{Cov(f(x),g(x,\sigma))^2}{Var(g(x,\sigma))} = \frac{(E[f(x)g(x,\sigma)] - E[f(x)]E[g(x,\sigma)])^2}{E[g(x,\sigma)^2] - E[g(x,\sigma)]^2} =$$

$$\frac{(\frac{1}{2}\int_{-1}^{1} f(x)g(x,\sigma)\,dx - (\frac{1}{2}\int_{-1}^{1} f(x)\,dx)(\frac{1}{2}\int_{-1}^{1} g(x,\sigma)\,dx))^2}{\frac{1}{2}\int_{-1}^{1} g^2(x,\sigma)\,dx - (\frac{1}{2}\int_{-1}^{1} g(x,\sigma)\,dx)^2} =$$

$$\frac{(\int_0^1 f(x)g(x,\sigma)\,dx - (\int_0^1 f(x)\,dx)(\int_0^1 g(x,\sigma)\,dx))^2}{\int_0^1 g^2(x,\sigma)\,dx - (\int_0^1 g(x,\sigma)\,dx)}$$

Intuitively, plotting $R(\sigma)$ should map out a peak near some region of $\sigma$ and focusing in on this interval should justly yield an approximated $\sigma$. Since the integral form of $R(\sigma)$ can be evaluated both by Mathematica's built in functions or by the pre-described method of MCI, the optimal $\sigma$ was evaluated using both methods for comparison (see next page for figure).

One can observe that both methods give similar values for an optimal $\sigma$. We will take $\sigma = 0.68376$ to reduce the variance and recalculate the initial integral in mind.

## 5. Results

Now that we have found the optimal value of $\sigma$, we define

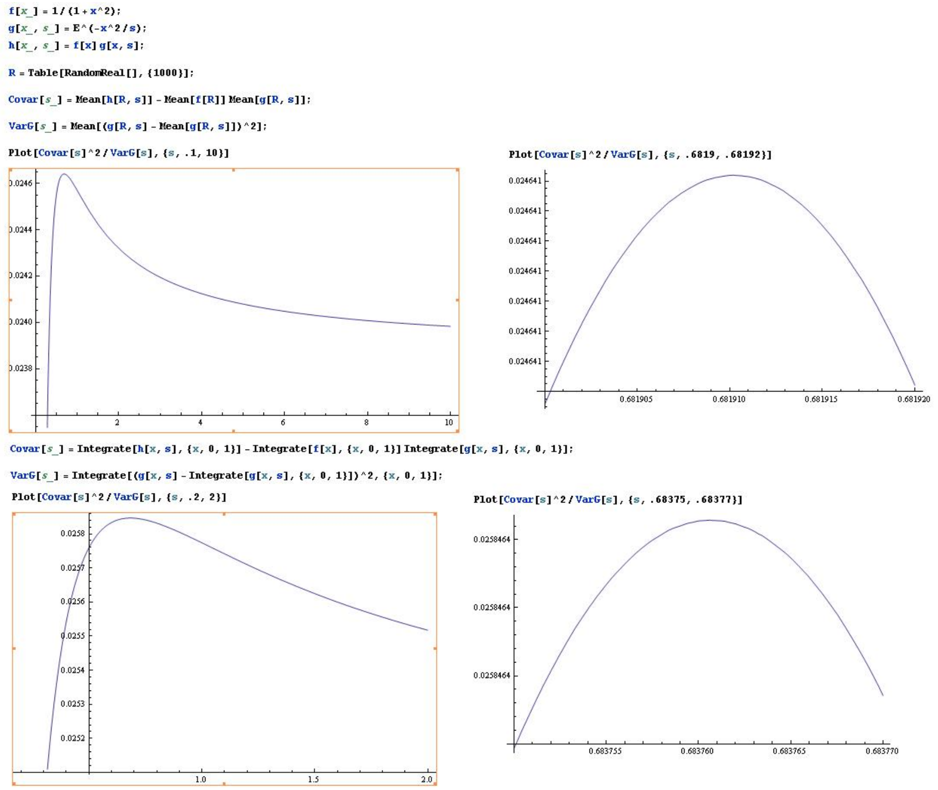
$$W = f(X) + a[g_\sigma(X) - \mu_g]$$

where $\sigma = 0.68376$. Next, we define $X$ to be a random variable uniformly distributed on the interval [0,1], that is $X \sim U[0,1]$. We then instruct Mathematica to repeatedly pick a random number between 0 and 1 to use as $X$ and then record the value of $W$ which is a function of $X$. Once this has been done several thousand times, the mean is taken as an estimate of $E[W] = E[f(X)]$. Recall that this is justified by the Law of Large Numbers explained previously. Finally, we multiply this estimate by 2 to get our estimate of the value of the desired definite integral. Doing this in Mathematica yields an estimate of 1.57179 which is fairly close to the known value of the definite integral $\frac{\pi}{2} \approx 1.5708\ldots$

Furthermore, we can double our estimate and obtain 3.14357, a fairly close estimate of $\pi$.

## 6. Acknowledgements

We would like to thank our professor, Dr. Walfredo Javier, and our graduate mentor, Ladorian Latin, as well as the VIGRE Summer program. We would also like to thank Dr. George Cochran, Dr. Robert Lax, Dr. Leonard Richardson, and Dr. Stephen Shipman for there insight into this study.



```
f[x_] = 1/(1+x^2);
g[x_, s_] = E^(-x^2/s);
h[x_, s_] = f[x] g[x, s];

R = Table[RandomReal[], {1000}];

Covar[s_] = Mean[h[R, s]] - Mean[f[R]] Mean[g[R, s]];

VarG[s_] = Mean[(g[R, s] - Mean[g[R, s]])^2];

Plot[Covar[s]^2/VarG[s], {s, .1, 10}]
```

```
Plot[Covar[s]^2/VarG[s], {s, .6819, .68192}]
```

```
Covar[s_] = Integrate[h[x, s], {x, 0, 1}] - Integrate[f[x], {x, 0, 1}] Integrate[g[x, s], {x, 0, 1}];

VarG[s_] = Integrate[(g[x, s] - Integrate[g[x, s], {x, 0, 1}])^2, {x, 0, 1}];

Plot[Covar[s]^2/VarG[s], {s, .2, 2}]
```

```
Plot[Covar[s]^2/VarG[s], {s, .68375, .68377}]
```

Figure 1. Finding the Optimal $\sigma$ Using MCI (top) and Built-In Integrate (bottom)

## References

[1] Ross, S., *Probability Models*, Elsevier, London, 2007.

[2] Ross, S., *A First Course In Probability*, Pearson Prentice Hall, Upper Saddle River, New Jersey, 1976.

[3] Parzen, E., *Modern Probability Theory,*, John Wiley Sons, New York, 1960.

Mathematics Department, Louisiana State University, Baton Rouge, Louisiana
*E-mail address*: dmiksa1@lsu.edu

Mathematics Department, Louisiana State University, Baton Rouge, Louisiana
*E-mail address*: pfacia1@lsu.edu

Mathematics Department, Louisiana State University, Baton Rouge, Louisiana
*E-mail address*: nick.cannady@gmail.com