Parabola Volume 56, Issue 3 (2020)

Monte Carlo and Other Integration Methods: How to Avoid Solving an Integral

Jennifer Lew[1]

1 Introduction

One of the most interesting courses offered at Palos Verdes Peninsula High School is Science Research. In this course, students connect with mentors in order to collaborate on independent research projects. In my case, I formed a mentor-mentee relationship with a Senior Utilities Engineer working at a government agency in my home state. As an introductory project, my mentor posed a simple problem involving engineering mechanics:

- You have a ruler that is secured in an upright position.
- The ruler has a length of L = 0.3m and a flexural rigidity of EI = 0.24044.
- A transversal force of P_T = 3.92N and a vertical force of P_V = 3N are applied to the tip of the ruler. These forces cause the ruler to deflect.
- Find the angle of deflection ϕ_o at the tip of the ruler.

This problem is depicted in Figure 1.

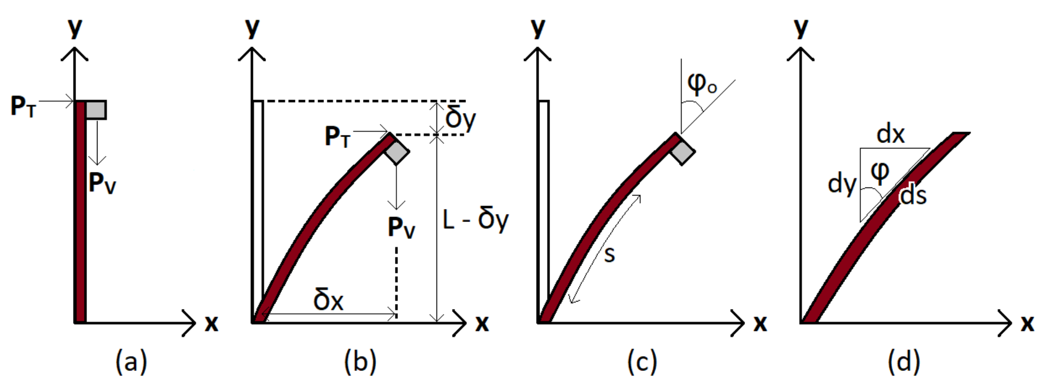

Figure 1: (a) Ruler with transversal and vertical forces, prior to deflection. (b) Ruler after deflection. (c) Arc length s and angle of deflection ϕ. The angle of deflection at the ruler tip is denoted as ϕ_o. (d) A small, approximately straight-line segment.

[1]Jennifer Lew is an 11th grader at Palos Verdes Peninsula High School in Rolling Hills Estates, California, USA.

My path toward finding ϕ_o started easily enough, and I was soon able to reduce the problem to a single integral. However, try as I might, I could not solve it. I surmised that the integral may have a known solution; but as a 16-year old, it was hard to know one way or the other (it's not as though an integral lends itself to an easy Google search). When I prompted my mentor for the solution, he responded cryptically, "you don't need to solve the integral, per se; you just need a numeric answer." After pondering his words for a while, I realized that I was pursuing an analytical solution where a numerical solution would suffice. I did a little digging and learned of a good numerical method for solving integrals: Monte Carlo Integration. In this paper, I will share how to use and optimize Monte Carlo Integration to solve a real-world problem.

2 My Integral

After working on the problem for some time, I was able to whittle the problem down to two equations:

$$1.35930 = \int_{0.59546}^{\frac{\pi}{2}} \frac{d\psi}{\sqrt{1 - k^2 \sin^2 \psi}} \tag{1}$$

$$k = \frac{1}{\sqrt{2}} \sqrt{1 + \sin(\phi_o - 0.65323)} \tag{2}$$

To solve for ϕ_o, you only need to guess k until (1) is satisfied, substitute the k value into (2), and then solve for ϕ_o. The integral in (1) looks simple enough, but the analytical solution evaded me. That's when my mentor suggested that I find a numerical solution instead. I imagine that, at this point, an experienced person would use a standard integration function (such as the quad function in Python). But for a 16-year old, finding a clearly-outlined strategy is really necessary for furthering mathematics education. So I cracked open my calculus book and researched the fundamentals of integration.

3 Methods to Solve an Integral

Riemann Sum

In calculus, you are taught that a definite integral can be solved by finding the area under the curve, where the curve represents the integrand. The area, in turn, can be calculated using a Left, Right, or Midpoint Riemann Sum. The Right Riemann Sum is shown here:

$$\int_a^b f(x)dx = \frac{b-a}{n} \sum_{i=1}^{n} f\left[a + (b-a)\frac{i}{n}\right] . \tag{3}$$

In a Riemann sum, the area under the curve $y = f(x)$ is divided into n rectangles, where each rectangle has a width of $(b - a)/n$. The area under the curve can then be found by summing the area of all the rectangles. To use a Riemann sum to solve the integral in (1), we express the integrand in (1) as a function:

$$f(\psi) = \frac{1}{\sqrt{1 - k^2 \sin^2 \psi}} . \tag{4}$$

To proceed, we need to guess a value for k. Let's guess k = 0.78956. This allows us to rewrite (4):

$$f(\psi) = \frac{1}{\sqrt{1 - 0.62341 \sin^2 \psi}} . \tag{5}$$

Using the function in (5), the integral in (1) can be put into the form of a Right Riemann Sum:

$$\int_{0.59546}^{\frac{\pi}{2}} \frac{d\psi}{\sqrt{1 - 0.62341 \sin^2 \psi}} = \frac{\frac{\pi}{2} - 0.59546}{n} \sum_{i=1}^{n} f\left[0.59546 + \left(\frac{\pi}{2} - 0.59546\right)\frac{i}{n}\right] . \tag{6}$$

The Right Riemann Sum in (6) is depicted pictorially in Figure 2 (where we set n = 10 to make the figure easier to draw). To estimate the integral in (1), I used a Right Riemann Sum with n = 100: the calculated value was 1.36021. I then used a Left Riemann Sum with n = 100: the calculated value was 1.35520. In both cases, our guess for k appears to satisfy (1). Subsequently, using (2), we find ϕ_o = 0.90262 or 51.72°.

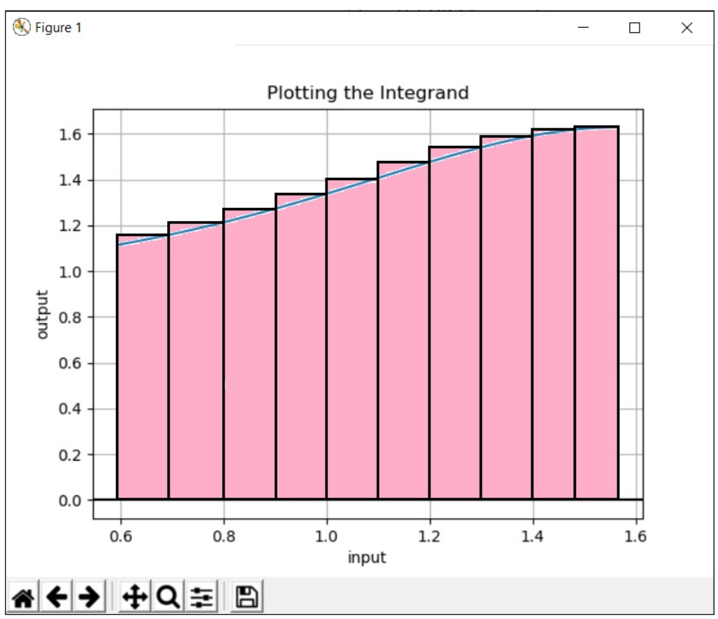

Figure 2: Solving the integral using a Right Riemann Sum with n = 10

Monte Carlo Integration

Another numerical method for solving an integral is the Monte Carlo Integration. This method is very similar to a Riemann sum; the only difference is that, instead of calculating the function at evenly-spaced input values (since every rectangle under the curve has the same width), Monte Carlo Integration calculates the function at random input values constrained between the limits of integration. An integral can be solved by Monte Carlo Integration as follows:

$$\int_a^b f(x)dx = \frac{b-a}{n} \sum_{i=1}^{n} f\left[a + (b-a)U_i\right] . \tag{7}$$

In (7), U_i is a random number between 0 and 1. To estimate the integral in (1), I used Monte Carlo Integration with n = 100: the calculated value was 1.36543. However, Monte Carlo Integration relies on random values, so running the calculation a second time will yield a different answer. Indeed, my second calculated value was 1.38356.

Monte Carlo Integration with Intervals

In Monte Carlo Integration with intervals, the function is calculated at n random input values constrained between the limits of integration. The limits of integration are first divided into evenly-sized intervals, and then the function is calculated at random input values constrained between the limits of each interval. For example, Figure 3 shows the limits of integration broken into five intervals. If we choose n = 100, then, within

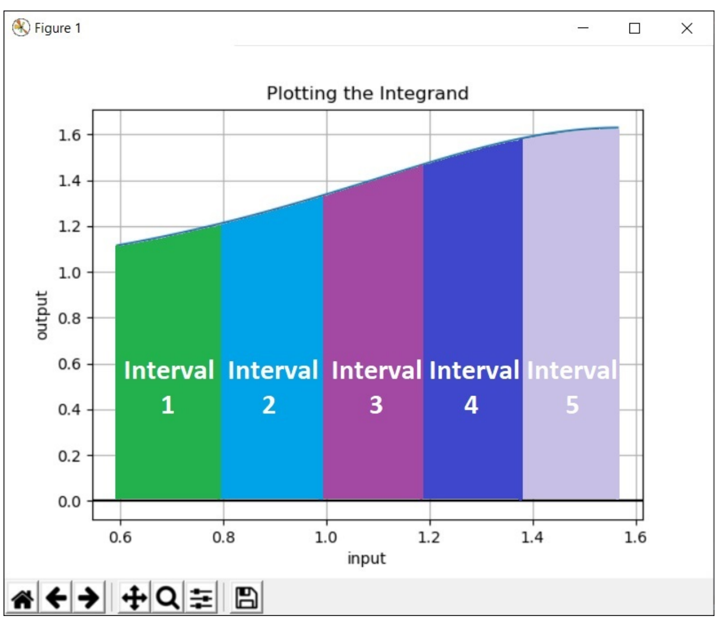

Figure 3: Monte Carlo Integration with 5 Intervals

each interval, the function is calculated at 20 random input values. To estimate the integral in (1), I implemented this method with n = 100 and 5 intervals: the calculated value was 1.35765. The calculation can be repeated using a fewer or greater number of intervals. Since n = 100, the most intervals you can use is 100.

Quad Function in SciPy

The SciPy library in Python has a function named "quad" that solves integrals. Using "quad", the integral in (1) was calculated to be 1.35771.

Analytical Solution

It turns out that the integral in (1) does have an analytical solution. The first step to finding the analytical solution is to break the integral in (1) into two integrals:

$$\int_{0.59546}^{\frac{\pi}{2}} \frac{d\psi}{\sqrt{1-k^2\sin^2\psi}} = \int_{0}^{\frac{\pi}{2}} \frac{d\psi}{\sqrt{1-k^2\sin^2\psi}} - \int_{0}^{0.59546} \frac{d\psi}{\sqrt{1-k^2\sin^2\psi}} \,. \tag{8}$$

On the right hand side of (8), the first term is the Complete Elliptic Integral of the First Kind and the second term is the Incomplete Elliptic Integral of the First Kind. With this in mind, we can rewrite (8) as:

$$\int_{0.59546}^{\frac{\pi}{2}} \frac{d\psi}{\sqrt{1-k^2\sin^2\psi}} = F_n\left(\frac{\pi}{2}, k, 0\right) - F_n\left(0.59546, k, 0\right)\,, \tag{9}$$

where

$$F_n\left(\frac{\pi}{2}, k, 0\right) = \frac{\pi}{2(2n+1)}\left[1 + 2\sum_{m=1}^{n}\left(1-\left[k\sin\left(\frac{m\pi}{2n+1}\right)\right]^2\right)^{-1/2}\right] \tag{10}$$

and
$$F_n\left(0.59546, k, 0\right) = \frac{1}{2n+1}\left[0.59546 + 2\sum_{m=1}^{n}\left(1-\left[k\sin\left(\frac{m\pi}{2n+1}\right)\right]^2\right)^{-1/2} \times \arctan\left(\tan 0.59546\left(1-\left[k\sin\left(\frac{m\pi}{2n+1}\right)\right]^2\right)^{1/2}\right)\right] \tag{11}$$

The constant n is the number of terms that you want to expand each elliptic integral. Using n = 50, I calculated the the integral in (1) to be 1.35930.

4 Accuracy of Each Integration Method

If we assume that the SciPy "quad" function is the most accurate, then Table 1 ranks each of the previously described integration methods in descending order of error. In Table 1, the column titled Result shows the calculated value. Notice that Monte Carlo Integration does not have a specified "Result". The reason is that Monte Carlo Integration uses random numbers; as such, every time you use this method you yield a slightly different result. The column titled Percent Error shows either the error or the average error. For each Monte Carlo Integration listed in the table, I ran 1000 trials and calculated the average error over all the trials. Based on Table 1, we see that Monte Carlo Integration without intervals is less accurate than a Riemann Sum. However, Monte Carlo Integration with at least 5 intervals is better than a Riemann Sum. The most interesting result comes from comparing a Riemann Sum with $n = 100$ to Monte Carlo Integration with 100 intervals and $n = 100$. The former calculates the function at 100 inputs that are evenly spaced apart; the latter calculates the function at 100 inputs chosen randomly from 100 intervals (one input is chosen per interval). Although both methods use 100 inputs, the Monte Carlo Integration with 100 intervals produces better results. It is not readily apparent to me why randomly choosing points within intervals is more accurate, but I surmise that doing so may more accurately capture the unique behavior of a function in a localized area.

Table 1: Error of Each Integration Method When Compared to SciPy Quad Function

Integration Method	Result	Percent Error
Monte Carlo Integration with $n = 100$	—	0.9326%
Monte Carlo Integration with 4 intervals and $n = 100$	—	0.2288%
Left Riemann Sum with $n = 100$	1.35520	0.1849%
Right Riemann Sum with $n = 100$	1.36021	0.1845%
Monte Carlo Integration with 5 intervals and $n = 100$	—	0.1818%
Analytical Solution (Elliptical Integrals)	1.35930	0.1171%
Monte Carlo Integration with 10 intervals and $n = 100$	—	0.0920%
Monte Carlo Integration with 20 intervals and $n = 100$	—	0.0443%
Monte Carlo Integration with 100 intervals and $n = 100$	—	0.0087%

5 Conclusion

The paper shows that Monte Carlo Integration with intervals can be more accurate than a Riemann Sum even when both methods use the same number of terms. For the particular integral in this paper, Monte Carlo Integration with at least 5 intervals showed greater accuracy. It is not readily apparent why this should be the case. Hopefully, further research into Monte Carlo methods will elucidate this curious result.

Thank you for reading my paper. If you are a secondary school student, then I hope you found my writing to be accessible and helpful. The goal of this paper is to translate some of the basic lessons from calculus to real-life problems. When I studied calculus, I never truly appreciated integrals and Riemann Sums. I thought that kind of material was introductory and only existed to lay the groundwork for more important mathematics. But now that I am solving real world problems, I see that the fundamentals of calculus - such as the simple definition of an integral - can be tremendous problem-solving tools.

Acknowledgements

I would like to thank my mentor Derek Fong, PE of the California Public Utilities Commission for providing challenging questions and subtle answers.