**Gaussian Quadrature**

An Approximation Method for Definite Integrals of Functions

A black and white photo of a person

Description automatically generated

Carl Friedrich Gauss(30 April 1777 – 23 February 1855).

Ralph Jordan Carlos Zapitan

Computational Physics, Fall 2019

University of Nevada, Las Vegas

**Abstract**

Programming languages are the fastest calculators in the world, but even them have their limits. Python, for example, can only do so much that it will eventually have trouble evaluating a function at too many points. In the case of numerical integration, which is what this topic is, speeding up our program is crucial. The more points we use, whether it is the Rectangle Rule, Simpson’s Rule, or the Trapezoid Rule, the better the approximation, but the slower our program will also be. Decrease the number of points, the program becomes much faster, but the accuracy of the approximation suffers dearly. Not even Simpson’s Rule can do both, but Gaussian Quadrature Rule has both of these abilities – it can obtain nearly exact values for just about any integrals using no more than five points. We can’t always find exact values for every integrals in the world, so we have to find ways to approximate, and Gaussian Quadrature is the next best thing.

**Background**

College students majoring in STEM usually begin their study of definite and indefinite integrals towards the end of their first semester Calculus. They are introduced to solving integrals of simple functions using analytical methods to find the exact area under a curve defined by a function.

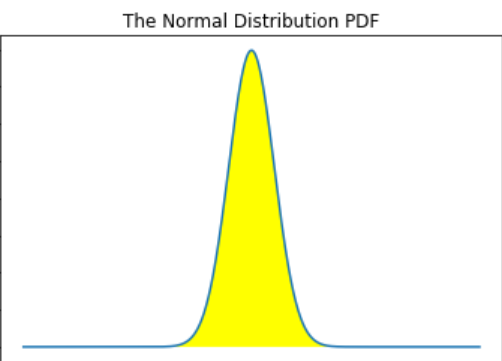
In general, solving integrals analytically is without question, the best method because exact solutions are obtained, and exact is always better. However, as students progress further into the course, they quickly realize that not all functions can be integrated analytically. In fact, only a handful of mathematical functions can be integrated analytically. In other words, while the derivative of just about any function can be obtained using differentiation, integration is not as easy - most integrals are impossible to solve by hand.

A close up of a logo

Description automatically generatedOne such function is the probability density function of the Normal Distribution in statistics given by

where is the mean or the expected value of the random variable and is the standard deviation of . One can try all day long to integrate this function by hand only to find out it is not doable.

This is the point where we turn to numeral analysis, and in the case of integration, numerical integration.

The graph of the Normal Distribution function.

**Numerical Analysis**

In Calculus, students also learn to compute definite integrals using numerical methods. For complex functions that can’t be integrated by hand, numerical approximation is the last resort, and may be the only way of finding areas under curves.

The most commonly used numerical integration methods taught in introductory Calculus courses are the Rectangle Method(also known as the Midpoint Rule), the Trapezoidal Rule, and Simpson’s Rule. As far as numerical integration is concern, these three are the most fundamental.

**The Midpoint Rule**

The Midpoint Rule, or the Rectangle Method, is the most basic of the three. Given an interval where the function of interest is to be integrated over, the interval is divided into subintervals of equal widths, , where . This is illustrated in the following diagram below.

A picture containing object, sky

Description automatically generated

The idea is to create rectangles under the curve and find the area of each individual rectangles using the formula , where and is the height where is either the left-endpoint of a rectangle for every rectangles under the curve or the right-endpoint of a rectangle for every rectangles under the curve, depending on whether the person performing the calculation wants to evaluate the height of each rectangle at the left-end point or the right-end point, and The total area under the curve over the interval is then approximated by summing up the areas of the rectangles. The general formula for this sum, or estimated area under the curve over the interval is

A picture containing object

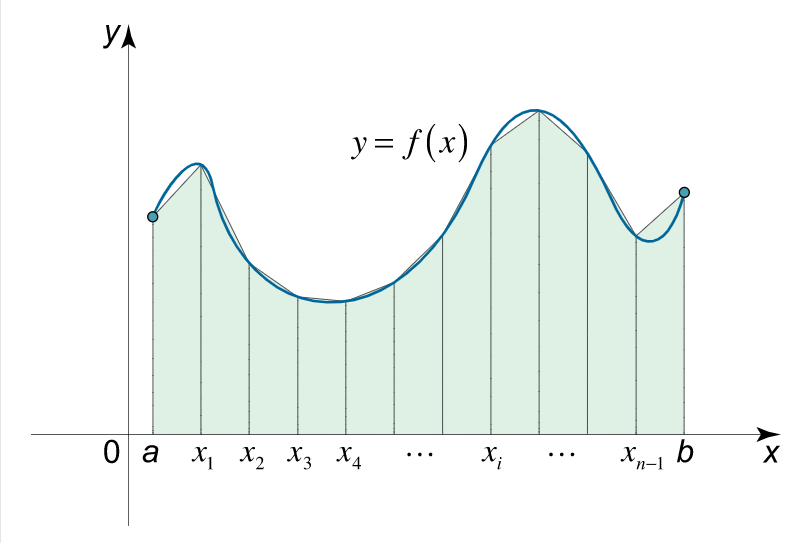
Description automatically generated

Notice that as the number of rectangles under the curve increases, that is, , the approximation becomes very accurate that the approximated area will be roughly equal to the actual area under the curve over the desired interval.

Because the Midpoint Rule would require us to use a large number of rectangles to get closer to the actual value of the definite integral, this method is very impractical and can be computationally expensive depending on how wide the interval we are trying to integrate our function over and the number of rectangles we have to use. A more convenient method is therefore preferred.

**The Trapezoidal Rule**

The Trapezoidal Rule is what comes after the Midpoint Rule. Like the Midpoint Rule, the interval of integration is also divided into subintervals with equal width . This is illustrated in the figure below.



Unlike the Midpoint Rule which breaks the interval of integration into equal rectangle widths , the Trapezoidal Rule uses trapezoids of equal widths . If we look closely at each diagram, we’ll notice that in the Midpoint Rule, for relatively small value of (the number of rectangles), there are empty spaces under the curve that are supposed to be filled by rectangles but are not and part of the rectangles are outside the curve – approximation with the Midpoint Rule is very unstable, it may either overestimate the true area or underestimate it if small values of is used.

The Trapezoidal Rule addresses this. The empty spaces under the curve are minimized. This results in a much more accurate estimation of the true area under the curve.

As with the Midpoint Rule, the width is given by and . The general rule for the Trapezoidal Rule is

.

The Trapezoidal Rule is superior to the Midpoint Rule and there is absolutely nothing wrong in overlooking the Midpoint Rule to use it all the time. However, as good as the Trapezoid Rule may appear, we can still obtain better approximations for definite integrals. Remember, as mentioned earlier, exact is always better.

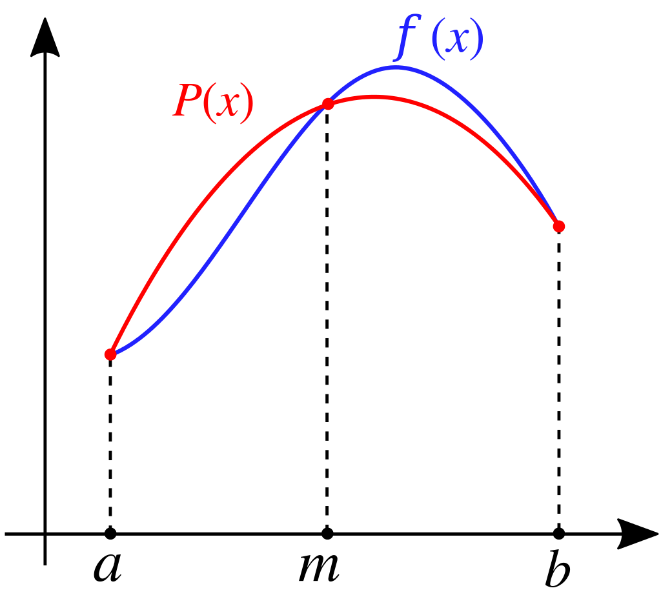
In Physics, the term ‘significant figure’ is more pronounced than it is in mathematics. While the Trapezoidal Rule is good enough in most cases to get reasonable approximation to definite integrals without the need to have to be accurate to many significant figures, there are cases when greater accuracy is a must. Physics professors will strictly enforce significant figure rules because what is mathematically correct may be wrong in terms Physics.

**Simpson’s Rule**

In certain situations where accuracy to more significant digits is required, Simpson’s Rule can generally meet the requirements for which the Trapezoid Rule is inadequate.

Simpson’s Rule always produces slightly more accurate approximation for definite integrals than the Trapezoid Rule for the same number of points , in the x-axis. Like the Trapezoid and Rectangle Rules, as , the approximation gets closer and closer to the true value of the integral.

Simpson’s Rule is similar to the Midpoint Rule and the Trapezoid Rule in a number of ways – the evaluation points s are evenly spaced. That is, for any two evaluation points and , the space between the two is given by , although in the case of Simpson’s Rule, must be an even non-negative integer, though this is not required for either of the other two methods. As shown in the figure below, Simpson’s Rule approximates the integral of some function by the quadratic interpolant .



The general rule for Simpson’s Rule is

Notice a pattern in the middle . The weights are alternating between 2 and 4. The figure above is demonstrates Simpson’s Rule with points . The figure below shows the method graphically with more points.

A close up of a map

Description automatically generated

The red lines do attempt to replicate the actual curve generated by the integrand in certain locations to give the corresponding area under the curve. As with the case with the Midpoint and Trapezoid Rules, the approximation of the total area under the curve in the interval becomes closer and closer to the actual area as , but Simpson’s Rule need not have as many points. Simpson’s Rule can match the accuracy of the Trapezoid Rule or even exceed it using way less points. This is crucial if we are trying to implement things in Python. The purpose of using as less points as possible while maintaining accuracy is to speed up our program; to approximate definite integrals in Python without Python choking on us. This is the main objective here: Use as less points as possible while still being accurate, so we can maximize the speed of our code. However, Simpson’s Rule, though better than the Trapezoid and the Midpoint Rules using less points, may still require a large number of points to obtain acceptable approximated integral values, and this slows down our computer. With these being said, is there a method that addresses this matter? It turns out that there is.

**Gaussian Quadrature**

The Gaussian Quadrature Rule is the last numerical integration rule that will be discussed and is the focal point of this entire report.

To reiterate things, the main goal is to allow Python to run at maximum speed when we are performing numerical integration, by using as less points of evaluations as possible, while still getting very accurate area under the curve of the function to be integrated. While Simpson’s Rule can’t quite achieve this remarkable feat, the Gaussian Quadrature rule easily does so.

Whereas the previous three other numerical integration methods all had one thing in common, which is the fact the evaluation points are evenly spaced apart, that is, the nodes are separated by equal widths , the Gaussian Quadrature Rule goes away from this concept – it evaluates definite integrals with great accuracy even when the nodes are unevenly spaced apart.

The Gaussian Quadrature Rule yields an exact result for polynomials of degree or less by a suitable choice of the nodes and weights for . One of the building blocks for the Gaussian Quadrature Rule is represented by the figure

A picture containing object

Description automatically generated

where are the nodes, are the corresponding weights. Understanding this rule is necessary in order to proceed to the general form of the Gaussian Quadrature Rule, which is given by

A picture containing object

Description automatically generated

From the above figure, it is easy to see that once we have the proper nodes and weights , we can readily approximate the integral on the left side of the equation, but how do we find them?

A picture containing text, map

Description automatically generatedYou may have heard of the Legendre-Polynomial. What does it have to do with the Gaussian Quadrature Rule? Well, it turns out, the two are tied together. Recall that there are infinitely many Legendre-Polynomial with different degrees. For example, the first two Legendre-Polynomial are and . In terms of sequence, this will look like , where is the degree of the Legendre-Polynomial. For example, means that the Legendre-Polynomial has degree 2, or that it is quadratic. To obtain the weights , we have to first get the nodes . The nodes are actually the roots of the nth Legendre-Polynomial, . Consider the graph of the Legendre-Polynomials with

As anticipated, no matter the degree of the Legendre-Polynomial, the nodes, which are the roots of the polynomials, will always be between -1 and 1. This is important because this is the only way the formula

A picture containing object

Description automatically generated

will be satisfied. Also, the number of roots, or nodes, is determined by the degree of the nth Legendre-Polynomial.

The process of the finding the nodes involves using Newton’s Method. Recall from Calculus that Newton’s Method is given by

A close up of a logo

Description automatically generated

Of course, we can’t print the nth Legendre-Polynomial by hand for many cases of n, so we have to develop an algorithm that allows our computer to recursively generate the Legendre-Polynomial for any n. The recursive rule that I am referring to is the following

A close up of a logo

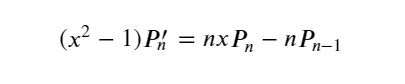
Description automatically generated

We have to feed to this algorithm the first two Legendre-Polynomials, which I said earlier to be and . Then, Python will obtain . Plug in and , we get . Python will keep doing this until it reaches the nth Legendre-Polynomial that we specify in a program. The next step is to obtain the corresponding weights of the nodes, . For this, we need to use the formula

A picture containing object

Description automatically generated

To get the second term in the square brackets in the denominator, we take the derivative of both sides of the previous equation, which comes out to be



In Python, the nodes of the nth Legendre-Polynomial must be stored in a list or an array. Then for each in that list, we evaluate this equation and at and store this in an array. Once all the s and the s have been stored in arrays, we are then ready to proceed to the last step of calculating the integral by using the equation

A picture containing object

Description automatically generated

Since our main objective is to maximize the speed of Python, we don’t want to use a list because it will require us to use loops, and we know that loops slow down our program. So we use arrays instead.

**Simpson’s Rule vs Gaussian Quadrature(Accuracy)**

The following plot demonstrates how the Gaussian Quadrature Rule is superior to Simpson’s Rule. We compare the two via the integral

A screenshot of a social media post

Description automatically generated

The x axis represents the number of points that the integral is evaluated at, while the y axis represents the error. The green plot represents the Gaussian Quadrature while the red represents Simpson’s Rule. The smaller the error, the better. Therefore, an error of zero means that the value of the integral is exact. Observing the graph, there is absolutely no contest between the Gaussian Quadrature and Simpson’s Rule. The error of the Gaussian Quadrature is very close to the y axis, indicative of its closeness to exactness of the area. Meanwhile, Simpson’s Rule does not seem to be doing a great job. Remember, one of the goals is to try to obtain values of definite integrals that are close to the exact value using as less points as possible so that the speed of our program is maximized. Looking at the graph, there is only one point where Simpson’s Rule did slightly better than the Gaussian Quadrature in terms of minimum error, but for every other points past that point, the Gaussian Quadrature completely destroys Simpson’s Rule. In fact, the value of the definite integral for every other points past this point is very close the actual value given by Scipy’s quad function, too close actually that they are almost exactly the same, just off by two or three decimals. This is not exclusive to this particular integral. The Gaussian Quadrature is guaranteed to outperform Simpson’s Rule in just about any numerical integration competition.

**References**

Dawkins, Paul. “Approximating Definite Integrals.” *Calculus II - Approximating Definite Integrals*, <http://tutorial.math.lamar.edu/Classes/CalcII/ApproximatingDefIntegrals.aspx>.

Svirin, A. (2019). Trapezoidal Rule. Retrieved November 24, 2019, from <https://www.math24.net/trapezoidal-rule/>.

qzhu2017, Q. Z. (n.d.). qzhu2017/ComputationalPhysics300. Retrieved November 24, 2019, from <https://github.com/qzhu2017/ComputationalPhysics300/blob/master/Lec_05_integral.ipynb>.

Wikipedia. (2019, November 26). Simpson's rule. Retrieved November 26, 2019, from <https://en.wikipedia.org/wiki/Simpson's_rule>.

Gaussian quadrature. (2019, November 26). Retrieved November 29, 2019, from <https://en.wikipedia.org/wiki/Gaussian_quadrature>.