# An Exploration of Gauss Quadrature

E. Nechanicky[1*]

December 11, 2023

## Abstract

This paper attempts an in-depth examination of Gauss quadrature. It details the theoretical foundations of Gauss quadrature, emphasizing its ability to achieve precise approximations with fewer points compared to methods like the trapezoidal or Simpson's rule. Central to the discussion is the method of selecting nodes and corresponding weights, particularly using Legendre polynomials for optimal approximation in fixed intervals.

The paper further explores practical aspects, including the computation of weights and nodes using root-finding algorithms like Newton-Raphson and the adaptation of Gauss quadrature for various functions and intervals, illustrated through examples of integrating non-polynomial functions. It also touches upon non-Legendre quadrature methods like Chebyshev Gaussian quadrature, demonstrating the difference in use case and accuracy with that of Legendre. Concluding remarks reflect on the value of Gauss quadrature in computational efficiency and accuracy, as well as its versatility in numerical analysis.

## 1 Introduction

### 1.1 Background

Numerical integration, a cornerstone of computational mathematics, is employed for calculating the integral of functions where analytical solutions are either too complex or impossible to find. Among various numerical integration techniques, Gauss quadrature stands out due to its precision and efficiency. This method is particularly advantageous when the function evaluations are costly or when dealing with integrals over a fixed interval. The efficacy of Gauss quadrature lies in its ability to provide highly accurate approximations using fewer points (nodes) compared to other methods like the trapezoidal or Simpson's rule. This paper delves into the theory of Gauss quadrature, its computational aspects, and its application in various complex integration scenarios.

## 2 Overview of Gauss Quadrature

### 2.1 Definition and Theory

Gauss quadrature, named after Carl Friedrich Gauss, is a method for approximating the definite integral of a function. The fundamental idea is to approximate the integral by a weighted sum of function values at specified points within the domain of integration. Mathematically, it can be expressed as:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{1}$$

where $a$ and $b$ are the limits of integration, $x_i$ are the 'nodes' or the points at which the function is evaluated, and $w_i$ are the weights assigned to each function value.

The essence of Gauss quadrature is its ability to select these nodes and weights optimally, such that the approximation is exact for polynomials of degree $2n-1$ or lower. This is a significant advantage over other numerical integration techniques that do not have this property.

### 2.2 Importance in Numerical Analysis

Gaussian quadrature is significant in numerical analysis due to its efficiency and accuracy. It is especially useful in scenarios where function evaluations are expensive or when high precision is required. In fields like physics, engineering, and finance, where generating precise numerical solutions is important, Gauss quadrature is a critical tool. Consequentially, due to its precision, Gauss quadrature often serves as a benchmark for evaluating the performance of other numerical integration methods.

One of the primary reasons for the efficiency of Gauss quadrature is its method of selecting nodes. Unlike methods like the trapezoidal rule, which use equally spaced nodes, Gauss quadrature is capable of selecting nodes that maximize the accuracy of the integral approximation for a given number of points. This optimal node selection is key to its high degree of precision. Additionally, Gauss quadrature requires significantly less function evaluations compared to other numerical integration techniques to achieve a similar level of accuracy. This is particularly beneficial when dealing with functions that are computationally intensive to evaluate.

## 3 Computing Weights and Nodes

### 3.1 Theoretical Background

The process of computing weights and nodes in Gauss quadrature involves finding the roots of basis polynomials. These roots determine the nodes, and the weights are calculated to provide the best possible approximation of the integral.

The selection of nodes and weights in Gauss quadrature is based on orthogonal polynomials. Two polynomials $P_n(x)$ and $P_m(x)$ are said to be orthogonal on an interval $[a,b]$ with respect to a weight function $w(x)$ if the integral of their products over that interval evaluates to 0 for all $n \neq m$. Mathematically, this is written as:

$$\int_a^b P_n(x) P_m(x) w(x) \, dx = 0, \; \forall \, n \neq m \tag{2}$$

For the standard Gauss quadrature, the orthogonal polynomials used are Legendre polynomials, which are orthogonal with respect to the weight function $w(x) = 1$ over the interval $[-1,1]$. The nodes are chosen as the roots of these polynomials, ensuring optimal placement within the interval. For this reason, this paper focuses primarily on Gaussian quadrature using Legendre polynomials as a basis. The specific implementation used computes the Legendre polynomials iteratively. Lether (1978) outlines the recurrence relationship used(3), and its improved efficiency, especially for higher degrees. Thus, the polynomial $P_n(x)$ is calculated using the recurrence relation:

$$P_k(x) = \frac{(2k-1)\, x P_{k-1}(x) - (k-1)\, P_{k-2}(x)}{k} \tag{3}$$

For the method of root-finding used in this paper, the derivatives of these polynomials will also be required. The derivative is given by:

$$P_k^{'}(x) = \frac{k\left[x P_k(x) - P_{k-1}(x)\right]}{x^2 - 1} \tag{4}$$

To extend Gauss quadrature to arbitrary intervals $[a,b]$, a linear transformation is applied to perform a change in variable. The resulting nodes and weights accurately approximate integrals over any fixed interval.

For Gauss-Legendre quadrature, this linear transformation takes the form:

$$x = \frac{b-a}{2} t + \frac{b+a}{2} \tag{5}$$

The weights $w_i$ are then scaled according to the length of the new interval, taking the form:

$$\hat{w}_i = \frac{b-a}{2} w_i \tag{6}$$

Therefore, the Gauss-Legendre quadrature formula for approximating the integral of a function $f(x)$ over the integral $[a,b]$ becomes:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t+\frac{b+a}{2}\right)dt \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}t_i+\frac{b+a}{2}\right) \tag{7}$$

## 3.2 Numerical Methods

The computation of nodes and weights in Gauss quadrature involves finding the roots of the associated orthogonal polynomials and then determining the weights that correspond to these nodes. For Legendre polynomials, various numerical methods can be employed.

### 3.2.1 Root-Finding Algorithms

Methods like the Newton-Raphson or the bisection method are commonly used to find the roots of Legendre polynomials. These roots become the nodes for the Gauss quadrature. The method used in this paper is Newton-Raphson, since it works well with the iterative method of calculating Legendre polynomials(3) and their derivatives(4).

**Algorithm 1** Newton-Raphson Root-Finding Algorithm

```
tolerance ← some small number
max_iterations ← some large number
current_guess ← initial_guess
for iteration = 1 to max_iterations do
    function_value ← EvaluateFunction(current_guess)
    derivative_value ← EvaluateDerivative(current_guess)
    if derivative_value ≈ 0 then
        break
    end if
    update ← function_value/derivative_value
    current_guess ← current_guess − update
    if |update| < tolerance then
        break
    end if
end for
return current_guess
```

Algorithm 1 outlines the implementation of Newton-Raphson for root finding used in this paper. The algorithm will start with an initial guess, a tolerance defining convergence criteria, and an iteration limit to stop if convergence is never achieved. At each iteration, the function this algorithm is executed over is evaluated at the current guess, and then the derivative is calculated at the current guess. The algorithm then updates the current guess from the calculated values. If the absolute value of the updated guess is less than the tolerance, then convergence is achieved and the guess is returned as the approximate root.

### 3.2.2 Computing Weights

Once the nodes are determined, the weights can be calculated using formulas derived from the properties of orthogonal polynomials. For Legendre polynomials, the weights are given by:

$$w_i = \frac{2}{(1-x_i^2)\,[P_n'(x_i)]^2} \tag{8}$$

where $P_n'(x_i)$ is the derivative of the nth Legrendre polynomial evaluated at the node $x_i$.

This formula is derived from the Christoffel Numbers Formula Adapted for Gauss-Legendre Weights (15.3.1 in Szegö (1959)):

$$\lambda_i = 2^{\alpha+\beta+1}\frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+1)\Gamma(n+\alpha+\beta+1)}\left(1-x_i^2\right)^{-1}\left\{P_n^{(\alpha,\beta)'}(x_i)\right\}^{-2} \tag{9}$$

where $i = 1,2,...,n$; $\alpha > -1$; and $\beta > -1$.

The Polynomial, $P_n^{(\alpha,\beta)'}(x_i)$ is a type of orthogonal polynomial known as a Jacobi polynomial. The parameters $\alpha$ and $\beta$ determine the weight function for which these polynomials are orthogonal. The general form of the weight function for Jacobi polynomials over the interval $[-1,1]$ is $(1-x)^{\alpha}(1+x)^{\beta}$ (See 4.1 in Szegö (1959)). From this, it is clear that Legendre polynomials are a special case of Jacobi polynomials where both $\alpha$ and $\beta$ are 0, simplifying the weight function to $(1-x)^0(1+x)^0 = 1$.

Thus, for Gauss-Legendre quadrature, specifically, $\alpha$ and $\beta$ are set to $\alpha = \beta = 0$, so the formula simplifies to:

$$\lambda_i = 2^1\frac{\Gamma(n+1)\Gamma(n+1)}{\Gamma(n+1)\Gamma(n+1)}\left(1-x_i^2\right)^{-1}\left\{P_n'(x_i)\right\}^{-2} \tag{10}$$

which further simplifies to:

$$\lambda_i = 2\left(1-x_i^2\right)^{-1}\left\{P_n'(x_i)\right\}^{-2} = \frac{2}{\left(1-x_i^2\right)\left\{P_n'(x_i)\right\}^2} \tag{11}$$

Both $\lambda_i$ and $w_i$ represent weights, and $x_i$ represent the nodes in the context of Gauss-Legendre quadrature. Thus, a direct translation between the two formulas is possible, allowing the weights of the Gauss-Legendre quadrature to be formulated as in (8).

# 4 Non-Legendre Quadrature

As previously mentioned, the selection of nodes and weights in Gauss quadrature is based on orthogonal polynomials. While Legendre polynomials are standard, other basis polynomials can be more effective for certain functions. One of these alternative basis polynomials is chebyshev. Chebyshev Gaussian quadrature uses Chebyshev polynomials, which are particularly effective for functions with endpoint singularities.(f is continuous but non-differentiable) The Chebyshev nodes and weights are given by:

$$\text{Nodes: } x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right), \quad \text{Weights: } w_i = \frac{\pi}{n} \tag{12}$$

There are several other Gaussian Quadrature variants that are fairly common. Considering these variants are not directly used within this paper, an in-depth discussion of the additional variants would not provide meaningful context. However, Table 1 provides an comparative overview of the main Gaussian quadrature variants. The usage of the Gauss-Chebyshev variant in the remainder of this paper should provide an understanding of how any one of these methods can be implemented in place of the standard Gauss-Legendre for weight functions where Gauss-Legendre is less optimal.

| Feature | Gauss-Legendre | Gauss-Chebyshev | Gauss-Laguerre | Gauss-Hermite |
|---|---|---|---|---|
| Orthogonal Polynomials | Legendre | Chebyshev | Laguerre | Hermite |
| Weight Function | 1 | $\frac{1}{\sqrt{1-x^2}}$ | $e^{-x}\ x > 0$ | $e^{-x^2}$ |
| Node Distribution | Uniformly within interval | Clustered near endpoints | Exponential growth along positive x-axis | Symmetric about origin |
| Node Calculation | Roots of Legendre polynomials | $\cos\left(\frac{(2i-1)\pi}{2n}\right)$ | Roots of Laguerre polynomials | Roots of Hermite polynomials |
| Weight Calculation | Varies with nodes | Equal weights | Varies with nodes | Varies with nodes |
| Ideal Use Cases | Smooth, bounded functions | Endpoint singularities | Functions decaying exponentially | Functions with Gaussian weight |
| Accuracy/Efficiency | Highly accurate for smooth functions | Efficient for specific endpoint behaviors | Suited for rapidly decaying functions | Effective for functions with Gaussian-like decay |
| Computation Complexity | Higher due to weights | Lower due to equal weights | Higher due to weights | Higher due to weights |
| Applicability | Broad range of functions | Functions with specific endpoint characteristics | Functions over $[0, \infty)$ | Functions over $(-\infty, \infty)$ with Gaussian weighting |

Table 1: Comparison of Different Gaussian Quadrature Methods

# 5 Application and Example

## 5.1 Practical Example

Example 1. Consider the integral $\int_{-1}^{1} e^x dx$. Using Gauss quadrature,

$$\int_{-1}^{1} e^x dx \approx \sum_{i=1}^{n} w_i e^{x_i} \tag{13}$$

where $w_i$ and $x_i$ are the weights and nodes, respectively.

Figure 1 shows the convergence rates for a variety of integral approximation methods with respect to the number of function evaluations performed. From this, it can be seen that Gauss-Legendre converges significantly faster than both the methods utilizing Simpson's Rule and Trapezoidal Rule. Notice that even within the Gaussian Quadrature convergence plot, there is clearly an optimal value for $n$ which will minimize the error. This can be seen as the Gauss-Legendre errors reach a minimum of around $10^{-15}$ for a certain $n$, after which the error begins to increase as $n$ increases.

Looking specifically at the difference in convergence between the Gauss-Legendre and Gauss-Chebyshev methods, the Gauss-Legendre method clearly outperforms Gauss-Chebyshev for this problem. In fact, it appears that even with very large $n$, the Gauss-Chebyshev implementation fails to converge at all. This is recognized by the fact that the Gauss-Chebyshev error is roughly constant for all $n$. This depicts the importance of understanding the specific use-cases for these alternative variants of Gaussian quadrature. Choosing the wrong basis polynomial for a specific problem can produce very poor results.

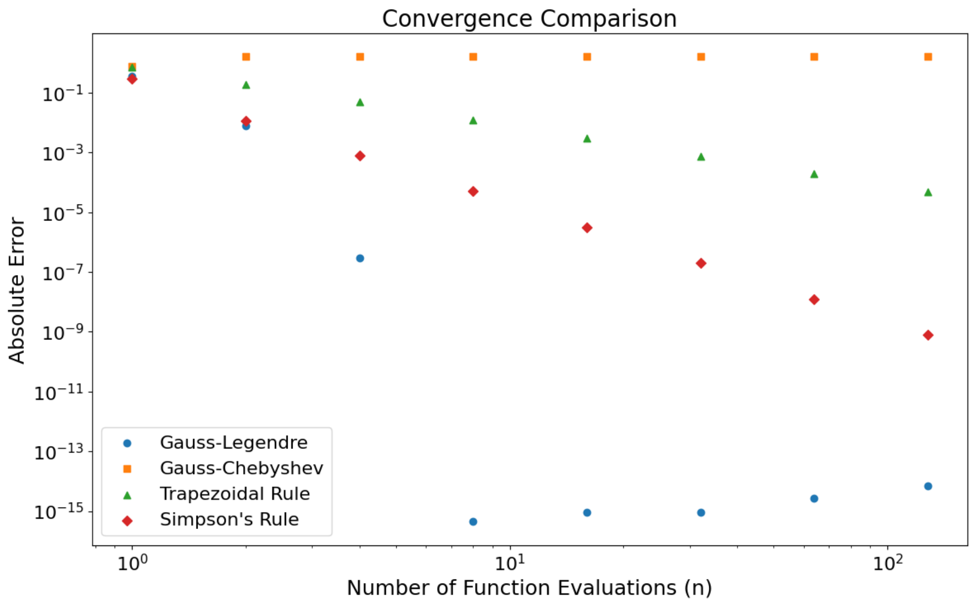


Figure 1: Integration convergence rate of various methods as n increases for $e^x$.

Example 2. Now consider the integral $\int_{-1}^{1} \frac{\cos(x)}{\sqrt{1-x^2}} dx$. Using Gauss quadrature,

$$\int_{-1}^{1} \frac{\cos(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n} \sum_{i=1}^{n} \cos(x_i) \tag{14}$$

where $x_i$ are the nodes, and the weights at each node are all $\frac{pi}{n}$, so the summation is multiplied by $\frac{pi}{n}$.

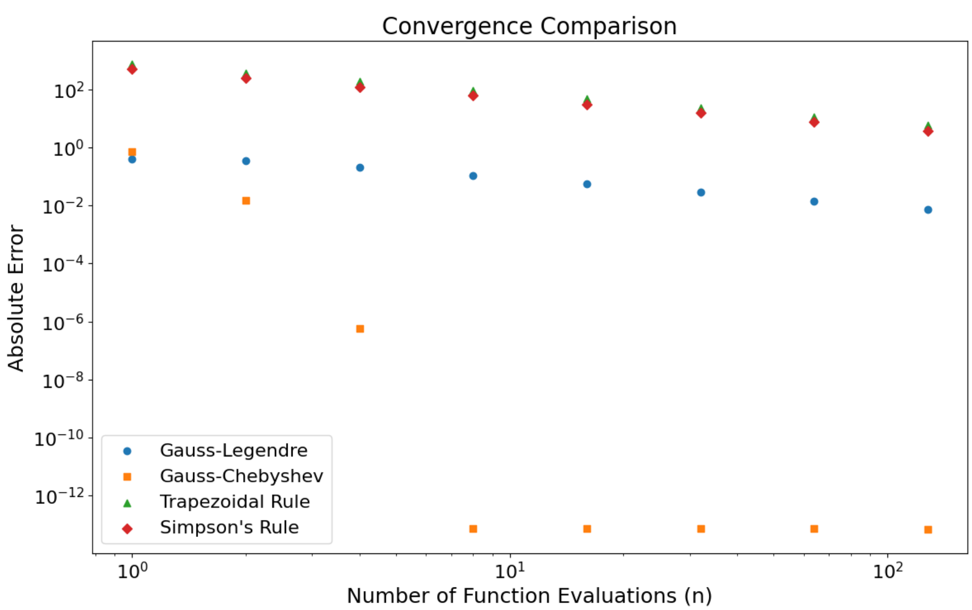


Figure 2: Integration convergence rate of various methods as n increases for $\frac{\cos(x)}{\sqrt{1-x^2}}$.

Figure 2 once more shows the convergence rates, however, this time the integral is approximated for equation 14. In this example, it the Gauss-Chebyshev method clearly outperforms Gauss-Legendre. This makes sense considering Gauss-Chebyshev is designed to work well for functions with a weight function of $w(x) = \frac{1}{\sqrt{1-x^2}}$, which is the case for this example. Additionally, in this example, the Gauss-Legendre method shows very small change for increases in $n$, suggesting that the method would likely fail to converge at a reasonable value of $n$. This further emphasizes the importance of choosing an appropriate basis polynomial for the problem being solved.

Furthermore, it is important to note that while the Gauss-Legendre method fails to outperform Gauss-Chebyshev, the Trapezoidal and Simpson's Rule methods are still outperformed by Gauss-Legendre. This further solidifies the concept of Gaussian quadrature being a more optimal approach to this kind of problem in general. Considering Gauss-Legendre is considered the standard method of Gaussian quadrature, it is probable that it will work to some degree for any function, even when there are more optimal variants of quadrature. This suggests that Gauss-Legendre will likely outperform methods like Trapezoidal rule and Simpson's Rule for any given function. However, that does not guarantee that any of those methods actually converge.

# 6 Conclusion

Gauss quadrature is a highly efficient and precise method for numerical integration, particularly for complex or computationally intensive functions. Its superiority over traditional methods like the trapezoidal or Simpson's rule is evident in its ability to achieve higher accuracy approximations with fewer nodes. This efficiency is largely attributed to the optimal selection of nodes and weights through the use of orthogonal polynomials. For most cases, the Legendre family of orthogonal polynomials is the optimal choice. However, where the Legendre polynomials come up short, other orthogonal polynomials, such as Chebyshev, can be used to adjust the procedure for the specific type of problem. The adaptability of Gauss quadrature to various functions and intervals, as demonstrated through examples in this paper, further underscores its versatility and applicability in a wide range of numerical analysis scenarios.

Moreover, the exploration of non-Legendre quadrature methods, like the Chebyshev Gaussian quadrature, highlights the importance of selecting appropriate basis polynomials based on the specific characteristics of the function being integrated. This choice can significantly impact the convergence and accuracy of the results, as shown in the comparative examples.

The computational aspects of Gauss quadrature, involving the calculation of weights and nodes using root-finding algorithms such as Newton-Raphson, are crucial for its practical implementation. The detailed discussion in this paper on these computational methods not only provides insight into the workings of Gauss quadrature but also offers a framework for its application in various numerical integration tasks.

Finally, this paper's in-depth examination of Gauss quadrature confirms its critical role in numerical analysis, offering computational efficiency, accuracy, and versatility. Its applicability in fields requiring high-precision numerical solutions, such as physics, engineering, and finance, makes it a critical tool in computational mathematics. Future research could explore the potential of Gauss quadrature in more complex and diverse application scenarios, as well as the development of more efficient algorithms for calculating weights and nodes.

## References

Lether FG (1978) On the construction of Gauss-Legendre quadrature rules. *Journal of Computational and Applied Mathematics* 4:47–52

Szegö G (1959) *Orthogonal Polynomials*. number v. 23 in American Mathematical Society colloquium publications. American Mathematical Society