International Journal of Applied Science and Engineering
Citation: *IJASE*: **12**(01): 107-124, June 2024
DOI: 10.30954/2322-0465.1.2024.9

REVIEW PAPER

NUMERICAL COMPUTATION

A Comprehensive Review of Numerical Integration Techniques and Error Analysis

A.K. Saha and Ch Santosh

Regional Institute of Education (NCERT), Bhubaneswar, Odisha, India

**Corresponding author: saha.ganit@gmail.com*

Received: 18 Mar., 2024 **Revised:** 26 May., 2024 **Accepted:** 02 Jun., 2024

ABSTRACT

This article provides a comprehensive overview of numerical integration, highlighting its significance in mathematical analysis and scientific computation. Beginning with a discussion on the importance, context, and necessity of numerical integration, it proceeds to explore various established methods such as the Newton-Cotes quadrature formula, the Lobatto integration method, and the Gaussian quadrature. The fundamental principles behind each technique are outlined, emphasizing the basic quadrature formulas that serve as the foundation for more advanced methods. Furthermore, an analysis of the errors associated with these numerical techniques is presented, underscoring the limitations and accuracy of each method. This study aims to equip readers with a clear understanding of the computational tools available for solving integrals that are otherwise difficult or impossible to compute analytically.

Keywords: Analytic function, Cauchy principal value, Filon's integral, quasi-exact method, error bound

1. Importance, context, and necessity of numerical integration

Analytically, an integral of the form

$$\mathcal{I}(u) = \int_a^b u(x)dx; \tag{1.1}$$

integrable in Riemann sense is exactly evaluated by using the "Fundamental Theorem of Integral Calculus". According to this theorem, if $\exists$ a differentiable function $U(x)$ such that

$$U'(x) = u(x);$$

then, in this case the integral

$$\int_a^b u(x)dx = U(b) - U(a).$$

Here $u(x)$ is known as the anti-derivative or the primitive of the integrand $u(x)$.

The method for assessing (1.1) seems to be straightforward and reliable, but it is not always a dependable method due to the following reasons:

How to cite this article: Saha, A.K. and Santosh, Ch. (2024). A Comprehensive Review of Numerical Integration Techniques and Error Analysis. *IJASE.*, **12**(01): 107-124.

Source of Support: None; **Conflict of Interest:** None

• We often come across integrals that are integrable in the Riemann sense, but finding their exact values analytically can be extremely challenging or even impossible. The following two examples may be considered in this category.

(a)

$$\int_0^1 e^{-x^2} dx \tag{1.2}$$

(b)

$$\int_0^1 \frac{ln\frac{1}{x}}{x} dx \tag{1.3}$$

• If the anti-derivative is determined, it appears in a critical form, which is very difficult to handle. One of the examples (Ref. Davis and Rabinowitz [?]) is given below

$$\int_0^x \frac{dt}{1+t^4} = \frac{1}{4\sqrt{2}} log \frac{x^2+\sqrt{2}x+1}{x^2-\sqrt{2}x+1} + \frac{1}{2\sqrt{2}} \left\{ tan^{-1}\left(\frac{x}{\sqrt{2}-x}\right) + tan^{-1}\left(\frac{x}{\sqrt{2}+x}\right) \right\}.$$

Though it seems that the exact value of the integral may be obtained from the above expression, in practice, it becomes an approximate value as it requires the evaluation of logarithmic and inverse tangent functions, which are practically evaluated to a certain degree of accuracy.

• In applications where the function is not given explicitly but the functional values are given, then an analytical method is not applicable.

• Moreover, finding the anti-derivatives of a great number of continuous functions, like $e^{-\sqrt{x}}$ and $ln(\sqrt{cosx})$, are not possible.

Here, an alternate approach is required, and numerical integration or mechanical quadrature came into being as a result. To use this method, we must first determine the precise value of the integral and then deal with its approximation.

The history of numerical integration is closely tied to the broader historical development of mathematics, computation, and the need for practical methods to handle integrals that lack closed-form solutions. Ancient mathematicians, including Greeks and Egyptians, used geometric methods to approximate areas under curves. Ancient Indian mathematical texts, such as the Sulba Sutras (dating back to around 800 BCE), contain geometric rules and methods for approximating areas. These early mathematical works, although not explicitly focused on numerical integration, laid the foundation for geometric considerations that are integral to the concept of quadrature. During the Middle Ages, Islamic mathematicians made contributions to numerical methods, including techniques for numerical integration. Notable figures include Alhazen and Ibn al-Haytham. The development of calculus by Newton and Leibniz in the 17th century provided a theoretical foundation for integration. However, practical numerical methods were still needed due to the limitations of analytical techniques. Leonhard Euler, in the 18th century, contributed to the development of numerical methods. He introduced Euler-Maclaurin summation, a technique that could be applied to numerical integration problems. Mathematicians like Euler and Gauss contributed to the development of numerical integration methods. Gauss, for instance, introduced Gaussian quadrature, a highly accurate method for specific classes of integrands. Computers emerged in the mid-20th century and brought about a significant transformation in numerical methods, including integration and numerical techniques that became widely used. In the 19th and early 20th centuries, mathematicians like Pafnuty Chebyshev and Charles Clenshaw developed quadrature rules for specific types of functions. Chebyshev-Gauss quadrature and Clenshaw-Curtis quadrature are examples of rules designed for functions defined on specific intervals. With the advent of computers in the mid-20th century, numerical integration methods gained new impetus. Traditional rules became widely used, and researchers started developing adaptive methods to automatically adjust the number of function evaluations in response to the integrand's behavior. The late 20th century saw the development of adaptive quadrature methods, where the integration interval is subdivided based on the function's behavior. Techniques like adaptive Simpson's rule and adaptive Gauss-Kronrod quadrature improve accuracy while minimizing the

number of function evaluations. In recent decades, research has focused on refining existing quadrature methods and developing specialized techniques for handling oscillatory integrals, singularities, and functions defined on irregular domains. High-performance computing has enabled the implementation of more sophisticated algorithms. The historical development of quadrature rules reflects the evolution of mathematical thought, and computational tools, and the increasing demand for accurate and efficient methods to tackle numerous integration problems encountered throughout diverse scientific and technical domains. Recent decades have seen the development of sophisticated algorithms and high-performance computing for numerical integration. Researchers continually refine and create new methods to address specific challenges posed by complex functions, high dimensions, and irregular geometries. Numerical integration has a rich history driven by the necessity to solve real-world problems where analytical solutions are often elusive or impractical. The interplay between theoretical developments and practical needs has shaped the evolution of numerical integration methods over centuries.

2. Methods of Numerical integration

To approximate a definite integral numerically, the following steps are required:

• **Approximation by Interpolation:**
In numerical computation, approximation by interpolation is a common technique used to estimate values within a given range based on known data points. These points are called nodes. The primary goal is to construct a function that closely matches the behavior of the underlying function or data set. This function is then used to estimate values at points between the known data points. Lagrange and Newton interpolation are two popular methods for interpolation.

• **Replacement of the Integral with Interpolating Polynomial:**
Replace the original integrand $u(x)$ with the interpolating polynomial over the same interval $[a, b]$. If $P(x)$ be the interpolationg polynomial, then the integral becomes $\int_a^b P(x)dx$.

• **Integration of Interpolating Polynomial:**
Evaluate the definite integral of the interpolating polynomial $P(x)$ in $[a, b]$. This involves determining the antiderivative of $P(x)$ and then evaluating it at the upper and lower limits of integration. The result provides a numerical approximation to the original definite integral.

These steps are particularly useful when the analytical integration of $u(x)$ is challenging, and using an interpolating polynomial simplifies the integration process. The accuracy of this method is quite sensitive to the interpolation technique and node count chosen. In regions where the function has rapid changes or oscillations, caution should be exercised, as interpolation may introduce errors. Additionally, using too few nodes may result in poor approximations, while using too many may increase computational costs without a proportional increase in accuracy.

3. Basic Quadrature Formula

Quadrature formulas are methods for approximating definite integrals, especially when an exact analytical solution is not feasible. The basic idea is to replace the integral with a weighted sum of function values at selected points. A basic quadrature formula typically involves evaluating the integrand at a set of points and using corresponding weights to approximate the integral.
Therefore, to find an approximate value of the integral

$$\mathcal{I}(u) = \int_a^b w(x_i)u(x_i)dx \tag{3.1}$$

where $w(x) > 0$ on $[a, b]$ called weight function.
we assume that $w(x)$ and $w(x)u(x)$ are integrable in Riemann sense on $[a, b]$. Now,

$$\mathcal{I}(u) = \int_a^b w(x)u(x)dx = \sum_{i=o}^{n} w(x_i)u(x_i) \tag{3.2}$$

where the nodes $x_i, w_i = w(x_i), i = 0(1)n$ dispersed inside the integration limits and $w_i, i = 0(1)n$ are called the weights of the quadrature formula. If $\mathcal{E}_n(u)$ is the error, then

$$\mathcal{E}_n(u) = \mathcal{I}(u) - \sum_{i=0}^{n} w_i u_i \tag{3.3}$$

4. Newton-Cotes Quadrature Formula

The general approach for deriving the Newton-Cotes quadrature formula is based on the Lagrangian interpolation technique. Suppose $u(x)$ be known for a set of $(n + 1)$ nodes: $x_0, x_1, x_2, \dots, x_n$. Now, if $P_n(x)$ is an interpolating polynomial of degree less than or equal to n, then

$$u(x_j) = P_n(x_j); j = 0,1,2, \dots, n,$$

and we can write

$$u(x) = P_n(x) + R_{n+1}(x).$$

Now integrating both sides within the limits a and b, we get

$$\mathcal{I}(u) = \int_a^b u(x)dx = I_n + \mathcal{E}_{n+1}, \tag{4.1}$$

where

$$I_n = \int_a^b P_n(x)dx$$

and

$$\mathcal{E}_{n+1} = \int_a^b R_{n+1}(x).$$

Taking the polynomial $P_n(x)$ as the Lagrange's form

$$P_n(x) = W(x) \sum_{r=0}^{n} \frac{u(x_r)}{(x-x_r)W'(x_r)}$$

where

$$W(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

and

$$W'(x_r) = \left[\frac{d}{dx} W(x)\right]_{x=x_r}$$

we get,

$$I_n = \sum_{r=0}^{n} u(x_r) H_r^{(n)}$$

where

$$H_r^{(n)} = \int_a^b \frac{W(x)dx}{(x-x_r)W'(x_r)}.$$

To evaluate $H_r^{(n)}$, we set, $x = x_0 + ht$.
Then, $H_r^{(n)} = (b - a)K_r^{(n)} = nhK_r^{(n)}$ where

$$K_r^{(n)} = \frac{(-1)^{n-r}}{n.r!(n-r)!} \int_0^n \frac{t(t-1)(t-2)\cdots(t-n)}{(t-r)} dt. \tag{4.2}$$

Therefore the Newton-Cotes formula is given by

$$\mathcal{I}(u) \approx I_n = (b - a)K_r^{(n)} \tag{4.3}$$

where $K_r^{(n)}$ is given in (4.2) and the error in this method is

$$\mathcal{E}_{n+1} = \int_a^b W(x) \frac{u^{n+1}(\xi)}{(n+1)!} dx, (a < \xi < b). \tag{4.4}$$

4.1 Trapezoidal Rule

The two-point Newton-Cotes formula is known as the trapezoidal rule. Here $n = 1$. Hence

$$\mathcal{I}(u) = \int_a^b u(x)dx = \sum_{i=0}^{n} w_i u(x_i);$$

$$\mathcal{I}(u) = w_0 u(x_0) + w_1 u(x_1),$$

where

$$w_0 = w_1 = \frac{h}{2}$$

Now

$$\begin{aligned} \mathcal{I}(u) &= w_0 u(x_0) + w_1 u(x_1) \\ &= \frac{h}{2}[u(x_0) + u(x_1)] \\ &= \frac{b-a}{2}[u(a) + u(b)] \end{aligned} \tag{4.5}$$

where

$$h = x_1 - x_0 = b - a$$

Error in Trapezoidal Rule

The error in the Trapezoidal rule is

$$\mathcal{E}_T(u) = \frac{-h^3}{12}u''(\xi); x_0 \leq \xi \leq x_n \tag{4.6}$$

4.2 Simpson's (1/3)rd Rule

The three-point Newton-Cotes quadrature formula is known as Simpson's (1/3)rd rule. For $n = 2$ we get

$$\mathcal{I}(u) = \int_a^b u(x)dx = \sum_{i=0}^{n=2} w_k u(x_k)$$

where

$$w_0 = \frac{h}{3} = w_2$$

and

$$w_1 = \frac{4h}{3}$$

Now

$$\begin{aligned} \mathcal{I}(u) &= w_0 u(x_0) + w_1 u(x_1) + w_2 u(x_2) \\ &= \frac{h}{3}[u_0 + 4u_1 + u_2]. \end{aligned}$$

Therefore

$$\mathcal{I}(u) = \frac{b-a}{6}[u(a) + 4u(\frac{a+b}{2}) + u(b)], \tag{4.7}$$

where

$$h = \frac{x_2 - x_0}{2} = \frac{b-a}{2}.$$

The Simson's (1/3)rd rule is obtained from the equation (4.7)

Error in Simson's (1/3)rd rule

The Error in Simson's (1/3)rd rule is

$$\mathcal{E}_{s\frac{1}{3}} = \frac{-h^5}{90}u^{iv}(\xi).$$

4.3 Simson's (3/8)th Rule

4-point Newton-Cotes formula is known as Simson's (3/8)th rule. For $n = 3$ we get

$$\mathcal{I}(u) = \int_a^b u(x)dx = \sum_{i=0}^{n=3} w_k u(x_k) = w_0 u(x_0) + w_1 u(x_1) + w_2 u(x_2) + w_3 u(x_3),$$

where

$$w_0 = \frac{3h}{8} = w_3$$

and

$$w_1 = \frac{9h}{8} = w_2$$

Now

$$\begin{aligned} \mathcal{I}(u) &= w_0 u(x_0) + w_1 u(x_1) + w_2 u(x_2) + w_3 u(x_3) \\ &= \frac{3h}{8}\left[u(a) + 3u(a + \frac{2h}{3}) + 3u\left(b - \frac{2h}{3}\right) + u(b)\right] \end{aligned} \tag{4.8}$$

where

$$h = \frac{x_3 - x_0}{3} = \frac{b-a}{3}$$

Equation (4.8) is called Simson's (3/8)th rule.

Error in Simson's (3/8)th Rule

The error associated with this method is

$$\mathcal{E}_{S\frac{3}{8}}(u) = \frac{-3h^5}{80} u^{iv}(\xi) \tag{4.9}$$

where $a \leq \xi \leq b$.

4.4 Boole's Rule

From Newton-Cotes method for $n = 4,$ we get the Boole's formula as

$$\int_{x_0}^{x_1} u(x)dx = \frac{2h}{45}[7u(x_0) + 32u(x_1) + 12u(x_2) + 32u(x_3) + 7u(x_4)] \tag{4.10}$$

In particular,

$$\int_{-1}^{1} u(x)dx = \frac{1}{45}\left[7\{u(-1) + u(1)\} + 32\left\{u\left(-\frac{1}{2}\right) + u\left(\frac{1}{2}\right)\right\} + 12u(0)\right], \tag{4.11}$$

where the required error is

$$\mathcal{E}_n(u) = \frac{-1}{3\times 7!} u^{vi}(\xi); \quad -1 < \xi < 1.$$

5. Lobatto Integration Method

The Lobatto integration formula with the interval $[-1,1]$ is

$$\int_{-1}^{1} u(x)dx = \frac{1}{6}\left[u(-1) + 5u\left(-\frac{1}{\sqrt{5}}\right) + 5u\left(\frac{1}{\sqrt{5}}\right) + u(1)\right]. \tag{5.1}$$

The error associated with this method is given by

$$\mathcal{E}_L(u) = \frac{-32}{526\times 6!} u^{vi}(\xi); \qquad -1 < \xi < 1 \tag{5.2}$$

6. Birkhoof-Young Quadrature Formula for Analytic Functions

The Birkhoof-Young (1950) [?] have derived the five-point formula $\mathcal{R}_B Y(u)$ as follows:

$$\int_{-1}^{1} u(z)dz = \frac{8}{5}u(0) + \frac{4}{15}\{u(1) + u(-1)\} - \frac{1}{15}\{u(i) + u(-i)\} + \mathcal{E}_B Y(u); \tag{6.1}$$

$$\begin{aligned}\mathcal{R}_B Y(u) &= \int_{\lambda_0-h}^{\lambda_0+h} u(z)dz \\ &= \frac{h}{3}\left[\frac{4}{3}\{u(\lambda_0+h)u(\lambda_0-h)\}+8u(\lambda_0)-\frac{1}{3}\{u(\lambda_0+ih)+u(\lambda_0-ih)\}\right]\end{aligned} \tag{6.2}$$

and

$$|\mathcal{E}_B Y(u)| \le \frac{1}{1890}|h|^7 \max_{z\in S}|u^{vi}(z)|. \tag{6.3}$$

Where S is the square with vertices defined as arguments in equation (6.3).

6.1 Modified Brikhoff-Young Formula

Let us consider an integral $\int_{-1}^{1} u(z)dz$ where $z \to u(z)$ is an analytic function in the square whose vertices are $1, -1, i, -i$. If we integrate Taylor expansion of u around $z = 0$ we obtain,

$$\int_{-1}^{1} u(z)dz = \sum_{n=0}^{\infty} \frac{u^{2n}(0)}{(2n+1)!}, \tag{6.4}$$

where

$$\begin{aligned}u^2(0) &= \frac{1}{2k^2}\{u(k)+u(-k)-u(ik)-u(-ik)\} \\ &\quad -2\left(\frac{k^2}{6!}u^{vi}(0)+\frac{k^8}{10!}u^{x}(0)+\frac{k^12}{14!}u^{xiv}(0)+\cdots\right)\end{aligned} \tag{6.5}$$

$$\begin{aligned}u^4(0) &= \frac{6}{k^4}\{u(k)+u(-k)+u(ik)+u(-ik)-4u(0)\} \\ &\quad -24\left(\frac{k^4}{8!}u^{viii}(0)+\frac{k^8}{12!}u^{xii}(0)\cdots\right)\end{aligned} \tag{6.6}$$

where $k, -k, ik, -ik$ $(k \le 1)$ are the vertices of the square in which we calculate the function u for an approximate computation of derivatives.

Applying (1.6.5) and (1.6.6) in (1.6.4), we obtain an integration rule

$$\begin{aligned}\int_{-1}^{1} u(z)dz &= 2\left(1-\frac{1}{5k^4}\right)u(0)+(16k^2+110k^4)\{u(k)+u(-k)\} \\ &\quad +\left(\frac{-1}{6k^2}+\frac{1}{10k^4}\right)\{u(ik)+u(-ik)\}+E,\end{aligned} \tag{6.7}$$

with the error term

$$E = \left(\frac{-2}{3}\frac{1}{6!}k^4+\frac{2}{7!}\right)u^{vi}(0)+\left(\frac{2}{9!}-\frac{2}{5\times 8!}k^4\right)u^{viii}(0)+\cdots$$

Putting $k = 1$, in equation (1.6.7), we get the $\mathcal{R}_B Y(u)$ formula (1.6.1) with a slightly changed error term as follows

$$\begin{aligned}\int_{-1}^{1} u(z)dz &= \frac{8}{5}u(0)+\frac{4}{15}\{u(1)+u(-1)\}-\frac{1}{15}\{u(i)+u(-i)\} \\ &\quad -\frac{1}{1890}u^{vi}(0)-\frac{1}{226800}u^{viii}(0)+\cdots\end{aligned} \tag{6.8}$$

It is to be noted that for

$$\frac{-1}{6k^2}+\frac{1}{10k^4} = 0,\ i.e.,\ for\ k = \sqrt{\frac{3}{5}},$$

when the coefficient of $\{u(ik)+u(-ik)\}$ vanishes, we obtain Gauss-Legendre-3 point formula $\mathcal{R}_{GL3}(u)$ as

$$\begin{aligned}\int_{-1}^{1} u(z)dz &= \frac{8}{9}u(0)+\frac{5}{9}\left[u\left(\sqrt{\frac{3}{5}}\right)+u\left(-\sqrt{\frac{3}{5}}\right)\right] \\ &\quad +\frac{1}{15750}u^{vi}(0)-\frac{1}{226800}u^{viii}(0)+\cdots\end{aligned} \tag{6.9}$$

7. Gaussian Quadrature

Gaussian quadrature is a numerical method for approximating definite integrals by choosing specific nodes (sample points) and weights. The key idea behind Gaussian quadrature is to use orthogonal polynomials, often the Legendre polynomials, to determine these nodes and weights. The result is a more accurate approximation of integrals compared to methods like the trapezoidal rule or Simpson's rule.

The general form of the Gaussian quadrature $(n+1)$-point formula for approximating

$$\mathcal{I}(u) = \int_a^b w(x)u(x)dx$$

over $[a,b]$ is:

$$\mathcal{I}(u) \approx \sum_{i=0}^{n} w_i u(x_i) \tag{7.1}$$

where x_i and w_i are selected in a manner that ensures the exactness of the formula for polynomials of degree $2n+1$ or less. The nodes are typically roots of orthogonal polynomials associated with the weight function on the interval $[a,b]$.

For Gaussian quadrature using Legendre polynomials, x_i and w_i are determined numerically. The Gaussian quadrature formula is usually presented in a standardized form:

$$\int_a^b w(x)u(x)dx.$$

If $w(x) = 1$, then the above method reduces to

$$\mathcal{I}(u) = \int_{-1}^{1} u(x)dx \approx \sum_{i=0}^{n} w_i u(x_i). \tag{7.2}$$

Rules for Basic Problems

The associated polynomial are Legendre polynomial $P_n(x)$ with n^{th} degree polynomial normalized to give $P(1) = 1$. The i^{th} Gauss nodes x_i is the i^{th} root of $P_n(x)$; its weight is given by Abramwoitz and Stegum.

7.1 Gauss-Legendre-2 Point Formula

Here $n = 1$. Then, the method becomes

$$\int_{-1}^{1} u(x)dx = \sum_{i=0}^{1} w_i u(x_i) = w_0 u(x_0) + w_1 u(x_1). \tag{7.3}$$

There are four unknowns in equation (1.7.3) and to determine the unknowns, it is assumed to be exact for polynomials of degrees up to 3.

So, $u(x) = x^i$ where $i = 0,1,2,3$. Now,

$$w_0 + w_1 = 2;$$

$$w_0 x_0 + w_1 x_1 = 0;$$

$$w_0 x_0^2 + w_1 x_1^2 = \frac{2}{3};$$

$$w_0 x_0^3 + w_1 x_1^3 = 0.$$

Solving the above system, we get

$$x_0 = \frac{1}{\sqrt{3}}; x_1 = \frac{-1}{\sqrt{3}}; w_0 = w_1 = 1.$$

The method (1.7.3) becomes

$$\int_{-1} 1u(x)dx = u\left(\frac{1}{\sqrt{3}}\right) + u\left(\frac{-1}{\sqrt{3}}\right) \tag{7.4}$$

The error associated with this method can be written as

$$\mathcal{E}_{GL2}(u) = \frac{u^{iv}(\xi)}{135}; \quad -1 < \xi < 1. \tag{7.5}$$

7.2 Gauss-Legendre-3 Point Formula

Here $n = 2$. Then the method becomes

$$\int_{-1}^{1} u(x)dx = \sum_{i=0}^{2} w_i u(x_i) \tag{7.6}$$

Solving the system of equations for $u(x) = x^i$ where $i = 0,1,2,3,4,5;$ we get,

$$x_0 = -\sqrt{\frac{3}{5}}, x_1 = 0, x_2 = \sqrt{\frac{3}{5}}$$

$$w_0 = \frac{5}{9}, w_1 = \frac{8}{9}, w_2 = \frac{5}{9}.$$

Hence the method (1.7.6) becomes

$$\int_{-1}^{1} u(x)dx = \frac{1}{9}\left[5u\left(-\sqrt{\frac{3}{5}}\right) + 8u(0) + 5u\left(\sqrt{\frac{3}{5}}\right)\right]. \tag{7.7}$$

The error term associated with the method (1.7.7) can be written as

$$\mathcal{E}_{GL3}(u) = \frac{8}{175\times 6!}u^{vi}(\xi); \quad -1 < \xi < 1. \tag{7.8}$$

The conceptual simplicity of the procedure of indeterminate coefficients may lead one to perceive a lack of interrelation among Gaussian quadrature rules. On the other hand, that's not correct. Rather, each Gaussian quadrature rule is connected with an underlying theory.

The points in Gauss quadrature are more than just the roots of certain orthogonal polynomials. Rather of using the approach of indeterminate coefficients, the weights can be more easily found by the Gram-Schmids orthogonalization process. Kopal [?], Krylov and Stroud [?], Hildebrand [?], and other renowned authors have covered the topic of alternative efficient approaches for identifying these unknowns.

Importantly, it should be mentioned that,

(i) with a specified number of nodes, the Gauss type of rules exhibit a maximal degree and produce results to maximum accuracy.

(ii) $w_j > 0; \forall j = 1,2,\ldots,n$ and

$$\sum_{j=1}^{n} w_j = \int_{-1}^{1} w(x)dx.$$

Hence, it is impossible for any coefficient to exceed the value of

$$\int_{-1}^{1} w(x)dx.$$

Consequently, we cannot have a situation in which a large coefficient creates large intermediate results causing cancellation when they are added.

(iii) Gaussian rules of different order have no nodes in common (except possibly the mid-point), so Gaussian rules are not progressive. Thus, estimating error using Gaussian rules of different order requires evaluating the integrand function at a full set of nodes of both rules.

Now, we will look at the Gauss quadrature rules with pre-assigned nodes.

7.2.1 Gauss-Lobatto Rule

This is a specialized form of Gaussian quadrature that includes the endpoints of the interval to enhance precision in the vicinity of these points. The rule is commonly used for approximating definite integrals on a finite interval, such as $[-1,1]$, where accurate integration near the endpoints is important. The rule over $[-1,1]$ is defined as follows:

$$\int_{-1}^{1} w(x)u(x)dx \approx \sum_{i=1}^{n+1} w_i u(x_i)$$

Here $w(x)$ is the weight function, $u(x)$ is the integrated function, x_i are the Lobatto points in the interval $[-1,1]$ and w_i are the weights associated with the Lobatto points x_i. The Lobatto points are chosen as the roots of the derivative of an orthogonal polynomial, often associated with Legendre polynomials. For the Gauss-Lobatto rule on $[-1,1]$, the Lobatto points are the roots of the derivative of the $(n+1)$-th degree Legendre polynomial. The weights w_i associated with the Lobatto points are determined using the following formula:

$$w_i = \frac{2}{(n+1)(n+2)[P_n(x_i)]^2}$$

where $P_n(x_i)$ represents the n-th degree Legendre polynomial evaluated at the Lobatto point x_i. The Gauss-Lobatto rule achieves a degree of precision of $(2n-1)$. This makes it particularly effective for approximating integrals involving functions with singularities at both endpoints.

The n-point Gauss-Legendre rule$(\mathcal{G}_n)$ and the $(n+1)$-point Gauss-Lobatto rule $(\mathcal{L}_{n+1})$ are of same precision $(2n-1)$. Combining these two rules we get the following rule.

$$\mathcal{R}_{GLc}(u) = \frac{1}{2n+1}[(n+1)\mathcal{G}_n + n\mathcal{L}_{n+1}].$$

The above-mentioned mixed quadrature rule is more efficient than the rules from which it is generated.

7.2.2 Gauss-Radau Rule

The rule is a special case of Gauss quadrature and is particularly useful when the integrand has a singularity at one end of the interval. For the interval $[a,b]$, the Gauss-Radau rule is expressed as:

$$\int_a^b w(x)u(x)dx \approx \sum_{i=1}^{n} w_i u(x_i) + w_{n+1}u(b)$$

The abscissas x_i for the interior points are typically chosen as the roots of an orthogonal polynomial associated with the weight function $w(x)$ over the interval (a,b). The weights w_i are then determined based on the derivative of the orthogonal polynomial at the corresponding abscissas.

The weight w_{n+1} associated with the endpoint b is often explicitly given in terms of known values. The Gauss-Radau rule achieves a degree of precision of $2n-1$. This high degree of precision makes it effective for approximating integrals involving functions with singularities at one endpoint.

7.2.3 Gauss-Chebyshev Quadrature Rule

In the Gauss-Chebyshev quadrature, the weight function

$$w(x) = \frac{1}{\sqrt{1-x^2}},$$

and the formula is given by

$$\int_{-1}^{1} \frac{u(x)}{\sqrt{1-x^2}} \approx \sum_{k=1}^{n} w_k u(x_k);$$

where the n-nodes $x_k = cos\left[\frac{(2k-1)\pi}{2n}\right], k = 1(1)n;$ are the zeros of the Chebyshev polynomial

$$\mathcal{T}_n(x) = cos[ncos^{-1}x].$$

Here all the weights are equal and are equal to $\frac{\pi}{n}$. The degree of precision of the n-point Gauss-Chebyshev quadrature rule is $(2n-1)$ which is the same as that of the n-point Gauss-Legendre quadrature rule.

7.2.4 Progressive Quadrature Rule

Any of the rules discussed above is not a **progressive rule.** Let $\mathcal{R}_m$ and $\mathcal{R}_n$ denote m-point and n-point rule respectively of the same type of rule with $n \geq m$ discussed above. If the integral is evaluated by both the rule $\mathcal{R}_m$ and $\mathcal{R}_n$ then the function evaluations done in the case of $\mathcal{R}_m$ can't be reused in $\mathcal{R}_n$ since all the nodes in $\mathcal{R}_m$ are different from those of $\mathcal{R}_n$ (except possibly at midpoint). Such rules are defined to be *non-progressive.* On the other hand, a rule is said to be progressive if the function evaluations by a m-point rule $\mathcal{R}_m$ are fully reutilized to compute the integral by the n-point rule $\mathcal{R}_n(n > m)$ of the same type with function evaluation at certain additional nodes of the rule $\mathcal{R}_n$. Clearly, a sequence of rules $\mathcal{R}_n, n = 1,2,\ldots$ is progressive if the nodes of $\mathcal{R}_{n-2}$ is a subset of the nodes of $\mathcal{R}_n$.

7.2.5 Example

Let $\mathcal{R}_{2n}^s; n = 2,4,8,\ldots$denote the compound Simpson's $(1/3)$rd rule for

$$\int_{-1}^{1} u(x)dx.$$

Then, the corresponding sets of points at which the function needs to be evaluated are given by

$$S_4 = \left\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\right\};$$

$$S_8 = \left\{-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\};$$

$$S_{16} = \left\{-1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{3}{8}, -\frac{1}{4}, -\frac{1}{8}, 0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1\right\}; \quad etc.$$

Clearly, $S_4 \subset S_8$ So, $\mathcal{R}_{2n}^s; n = 2,4,8$ is a progressive rule according to the definition. Thus, in the case of progressive quadrature formulas:

(i) Each formula reuses the nodes or points of function evaluation of the earlier formula reducing the volume of work and consequently cost and time of computation. The Kronrod quadrature rule, which will be covered later, is also motivated by the need to prevent this extra labor.

(ii) We can increase the number of points (nodes) n to get higher accuracy. Either way, efficiency is increased if subsequent rules are progressive, requiring fewer new integrand assessments in order to produce an approximation with a better degree of accuracy.

(iii) Gauss type rules are not progressive as the nodes of n-point formulas are all different from a $m(\neq n)$ point formula since they are the zeros of certain n^{th} degree orthogonal polynomials.

(iv) Sometimes, it is also not possible to make a decision on a particular formula to use or not to use to achieve accuracy to a desired figure although the error term of the rule is available in analytical form. In that situation, one can resort to a progressive formula to evaluate an integral and to empirically measure the error from two consecutive approximations obtained. If this error continuously approaches zero and at any stage, the computed error satisfies the terminating condition such as

$$|\mathcal{R}_{2n-2}^s - \mathcal{R}_{2n}^s| < 0.5 \times 10^{-n};$$

then one can confidently take the numerical output obtained at the step where the terminating condition is satisfied as the true value of the integral. This method saves time, work, and cost of computing as well since each step of evaluation reused the function values at its earlier step.

7.2.6 Adaptive Quadrature Rule

This is a numerical integration technique that dynamically adjusts the step size to achieve a desired

level of accuracy. Unlike fixed-step-size methods, adaptive quadrature adapts to the local behavior of the integrand, using smaller steps in regions where the function varies rapidly or has singularities and larger steps in smoother regions. This adaptability makes adaptive quadrature particularly effective for accurately integrating functions with varying characteristics.
Let us assume that there are strong variations in the orders of magnitude of the integrand $u(x)$ in distinct regions of the integration interval $[a, b]$. So, different step sizes should be used in various regions of the integration interval. If we write

$$\int_a^b = \int_a^{\alpha_1} + \int_{\alpha_1}^{\alpha_2} + \cdots + \int_{\alpha_n}^b$$

then It is possible to consider the integrals on the right-hand side as separate subproblems. Adaptive quadrature methods employ automatic adjustment of step sizes to ensure that the approximation meets a predetermined error tolerance:

$$|\mathcal{I} - \int_a^b u(x)dx| \leq \epsilon.$$

Adaptive quadrature is widely used in scientific computing and numerical analysis because of its ability to efficiently handle integrands with complex behavior. It is especially valuable when dealing with functions that have singularities, rapid oscillations, or other challenging characteristics, as it automatically adjusts the step size to capture the relevant features accurately.

8. Error analysis

The solutions or values obtained by numerical analysis should not be precise analytical outcomes. Given this challenge, we ought to be aware of how to properly build or apply techniques or algorithms so that the results are obtained concurrently. Error analysis in numerical quadrature, or numerical integration, involves studying the difference between the approximate numerical result obtained by a quadrature method and the exact value of the integral. Understanding and quantifying this error is crucial for assessing the reliability and accuracy of numerical integration methods.
Numerous techniques for solving integrals were covered in the previous section. Since a method's accuracy is ultimately determined by examining the deviation between an integral's exact value and the value produced by different rules, we can say that a method is more accurate if that difference is significantly smaller in digits than other methods. In this part, we felt it would be useful to talk about various methods for determining the error bounds of numerical quadratures for future research. After this section, some more techniques or guidelines for approximating definite integrals will be revisited from the perspective of correctness. Finite expressions of the kind

$$\mathcal{R}_k(u) = \sum_{i=0}^n w_i u(x_i); \quad (8.1$$

are often computed in numerical integration, and the obtained result is treated as an approximation to the definite integral

$$\mathcal{I}(u) = \int_{-1}^1 u(x)dx. \quad (8.2$$

In numerical methods, truncation error refers to the inaccuracy that arises while approximating an integral using a finite number of terms. The quality of the procedure and the accuracy of the approximation are interconnected. The quantification of truncation error is commonly represented by the leading term in the series expansion of the error, which is dependent on either the step size or the number of nodes employed in the quadrature method.
The truncation error, denoted by $\mathcal{E}_n(u)$ provided by:

$$\mathcal{E}_n(u) = \mathcal{I}(u) - \mathcal{R}_n(u); \tag{8.3}$$

Nevertheless, because the numbers x_i, w_i *and* $u(x_i)$ have a finite representation, round-off error occurs during processing on a machine. Occasionally, a tiny round-off error causes noise to be produced. Nonetheless, the round-off error is minuscule when (8.1) has few terms. Standard texts such as Hildebrand, Davis and Rabinowitz, Hamming, etc., provide a fairly comprehensive treatment of the various causes of error and the means of controlling them.

Here we quickly go over some methods employed for the analysis of truncation error in numerical integration, which is all that we'll be focusing on in our treatment.

If d is the degree of precision of the rule $\mathcal{R}_n(u)$ and $[-1,1]$ contain the nodes x_i, then $\mathcal{E}_n(u)$ is given by:

$$\begin{aligned} \mathcal{E}_n(u) \quad &= \frac{1}{(d+1)!}\int_{-1}^{1} x^{d+1}u^{(d+1)}(\xi)dx \\ &\quad -\sum_{i=1}^{n} w_i x_i^{d+1}u^{(d+1)}(\xi_i); \end{aligned} \tag{8.4}$$

where $-1 < \xi_i < 1,\ -1 < \xi < 1$ and the function $u(x)$ is continuously differentiable $(d+1)$ times in $[-1,1]$. The error $\mathcal{E}_n(u)$, as shown in reference (8.4), is not appropriate for handling due to the inclusion of unknown points $\xi_1, \xi_2, \ldots, \xi_n$ and higher order derivatives of the integrand $u(x)$ when the value of d is large. However, when considering the error term as a functional, it can be more readily represented using the Peano Kernel theorem (Davis and Rabinowitz).

$$\mathcal{E}_n(u) = \frac{1}{d!}\int_{-1}^{1} u^{(d+1)}(s)\mathcal{E}_n^x(x-t)_+^d ds; \tag{8.5}$$

where $\mathcal{E}_n^x$ represents to $\mathcal{E}_n$ as functional operates on x-variable and

$$(x-t)_+^d = \begin{cases} (x-t)^d, & x \geq t \\ 0, & x < t \end{cases} \tag{8.6}$$

If the quantity $\mathcal{E}_n^x(x-t)_+^d$ does not change sign for $-1 \leq x \leq 1$ we have in particular

$$\mathcal{E}_n(u) = \frac{u^{(d+1)}(\xi)}{(d+1)!}\mathcal{E}_n(x^{d+1}), \xi \in (-1,1). \tag{8.7}$$

The error term in (8.5) or (8.7) does not have the several unknown points ξ_i as it does in (8.4), but it does have the integrand's derivative and the one unknown ξ_i, which makes it unfavorable. Theoretical and practical approaches to finding the boundaries for the error $|\mathcal{E}_n(u)|$ that are independent of the integrand's derivatives have been the subject of extensive study using complex analysis and functional analysis. Sharpness at the limit of $|\mathcal{E}_n(u)|$ would be ideal.

8.1 Error bounds using complex analysis technique

Error analysis using complex analysis techniques can be a powerful approach to derive precise estimates of the error in numerical integration. In particular, complex analysis allows for the exploration of the behavior of the integrand in the complex plane, providing insights into the convergence properties of the numerical method. Here are some techniques commonly used in complex analysis for error bounds in numerical integration:

Contour Integration:

Complex analysis often involves the use of contour integration to evaluate integrals. Contour integration allows one to deform the integration path in the complex plane, potentially avoiding singularities or choosing paths that simplify the computation.

Residue Theorem:
This is a very effective tool in complex analysis that relates contour integrals to the residues of a function within a closed curve. When applying the residue theorem to numerical integration, one can estimate the error by analyzing the behavior of the residues near singular points.

Jordan's Lemma:
Jordan's lemma is used to analyze the behavior of integrals along a semicircular contour in the upper or lower half-plane. It can be employed to estimate the error when integrating along paths that circumvent singularities.

Analytic Continuation:
Complex analysis allows for the study of functions in the complex plane and their analytic continuation. By understanding the behavior of a function in the complex plane, one can make informed decisions about the integration path and estimate the error.

Mapping Techniques:
Complex mapping techniques involve transforming the integration contour or the integrand itself using conformal mappings. This can simplify the integration process and aid in error analysis.

Saddle Point Method:
The saddle point method is used to approximate integrals by locating critical points of the integrand. In the context of numerical integration, this method can provide insights into the dominant contributions to the integral and help estimate errors.

Cauchy's Integral Formula:
It expresses the value of a function on a closed curve in terms of its values on the curve's interior. This formula is useful for understanding the behavior of functions within a contour and can be applied to error analysis.
These complex analysis techniques can be particularly beneficial when dealing with integrals that involve oscillatory behavior, singularities, or complex-valued functions. The insights gained from complex analysis can lead to more accurate error estimates and help guide the selection of appropriate numerical integration methods.
It's important to note that while complex analysis provides powerful tools, its application to error analysis in numerical integration depends on the specific characteristics of the integrand and the problem at hand.

8.2 Asymptotic error estimate

Asymptotic error estimates are analytical expressions that provide information about how the error of a numerical approximation decreases as the resolution (such as the number of subintervals or data points) increases towards infinity. These estimates are often expressed in terms of mathematical functions that describe the behavior of the error as the problem size grows.
For numerical integration methods, such as the trapezoidal rule, Simpson's rule, or more advanced methods like Gaussian quadrature, asymptotic error estimates help analyze the convergence behavior of the method as the number of intervals increases. Typically, these estimates involve the step size h or the number of intervals n. If $\mathcal{E}_n(u)$ is the exact error and $\hat{\mathcal{E}}_n(u)$ is an estimate of $\mathcal{E}_n(u)$ then $\hat{\mathcal{E}}_n(u)$ is said to be an asymptotic estimate for $\mathcal{E}_n(u)$ if

$$\lim_{n\to\infty} \frac{\mathcal{E}_n(u)}{\mathcal{E}_n(u)} = 1 \tag{8.8}$$

$$\Leftrightarrow \frac{\mathcal{E}_n(u)-\hat{\mathcal{E}}_n(u)}{\mathcal{E}_n(u)} = o(1).$$

For example the exact error in compound Simpson's rule:

$$\int_{-1}^{1} u(x)dx \approx \frac{h}{3}[u(-1)+u(1)+4\{u_1+u_3+\ldots+u_{n-1}\}+2\{u_2+u_4+\ldots+u_{n-2}\}]; \tag{8.9}$$

where

$$\begin{cases} n \geq 2 \text{and even}, \\ h = \frac{2}{n}, \\ u_j = u(-1+jh), \quad j = 1,2,\ldots,n-1; \end{cases}$$

is given by

$$\begin{aligned} \mathcal{E}_n(u) &= -\frac{h^5}{90}\sum_{j=1}^{n/2} u^{(iv)}(\eta_j) \\ &= -\frac{h^4}{90}u^{(iv)}(\eta); \end{aligned}$$

where all $\eta_j \in (-1,1)$ and $\eta \in (-1,1)$. An asymptotic error estimate of the compound Simpson's rule given in (8.9) is known to be

$$\hat{\mathcal{E}}_n(u) = -\frac{h^4}{180}[U''(1)-U'''(-1)].$$

In his explanation of the asymptotic error formulas, Atkinson also notes their applications. The authors Asheim and Huybrechs, Barnhill Chawla and Jain and Chawla have also given asymptotic error estimates for Gauss quadrature formulas. One thing we can see from the error terms is that the length of the integration interval has a direct correlation with the error. Additionally, by analysing the error terms, we may determine the precision of the quadrature formula for a specific set of polynomials.

8.2.1 Compound and composite rule

For numerical approximations of definite integrals, we have already covered some basic quadrature rules in section (2). Section (8) once again covers the error terms linked to these fundamental laws, this time utilizing various mathematical strategies to establish the error bound. Because it's important to have regulations that are accurate, it's clear from comparing and contrasting these two parts that there are some problems with putting these rules into effect.

Some fundamental quadrature methods for numerical approximations of definite integrals were covered earlier in section (2). The section (8) previously addressed the topic of the error terms linked to these basic rules, which are determined using various mathematical procedures. Since a rule's accuracy is important, we can see by comparing and contrasting these two parts that there are several challenges with putting these rules into effect. Which are

1. Basic rules like trapezoidal or Simpson's rule often have limited accuracy on larger intervals. Dividing the interval into smaller segments and applying the rule multiple times can significantly improve accuracy.
2. Once again, as we saw with the 9-point Newton-Cote's rule of precision nine, round-off errors would increase if some quadrature weights were negative for a big number of nodes acting as abscissas in an integration interval.

Hence, in case the interval of integration is sufficiently broad, an alternative approach becomes necessary for enhancing the accuracy in an approximation of the integral by basic quadrature rules.

It is well known that, all basic quadrature formulas are very simple to use and of lower precisions. Further, each of them approximates an integral efficiently, when the integral is defined with a smaller interval of integration. Therefore, for large intervals to achieve reasonable accuracy by employing the basic quadrature rules; the rules are executed over smaller subintervals of the intervals of integration by virtue of the additive rule

$$\mathcal{I}(u) = \int_a^b u(x)dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} u(x)dx;$$

where

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b.$$

The concept of compound rules is derived from this, and they are described below. By partitioning $[a, b]$ into n equal parts, we can apply a consistent and straightforward rule $\mathcal{R}_n(u)$ to each of these n-subintervals and get a series of approximations. The approximate value of the integral $\mathcal{I}(u)$ is then obtained by adding these resultant partial approximations.

It is recommended to employ the compound rule instead of the basic rule that applies to the entire interval, commonly referred to as holistic rules by Mustard, to get a pretty accurate result of an integral (Lyness). Approximating integrals of functions with specific kinds of singularities is another use of the compound rules. According to the theorem presented in Davis and Rabinowitz, compound rules produce more precise outcomes as compared to their holistic counterparts.

Furthermore, it can be observed that approximations derived from a n-point compound rule of these fundamental rules tend to approach the true value of the integral as $n \to \infty$ because compound rules associated with numerous basic rules can be represented as Riemann sums. On the other hand, in terms of practicality, the round-off error only allows us to obtain approximations that are accurate up to specific decimal places.

The convergence of integral values for the rectangle rule, trapezoidal rule, and compound form of Simpson's(1/3)rd rule occurs as the number of sub-intervals n approaches infinity, as these integrals are all Riemann sums. Nevertheless, different regulations lead to different rates of convergence.

The compound Simpson's (1/3)rd rule exhibits convergence at a rate of $\left(\frac{1}{n^4}\right)$, whereas the compound trapezoidal rule demonstrates convergence at a rate of $\left(\frac{1}{n^2}\right)$. For the given value of $n = 10$, it is expected that the compound Simpson's rule will provide an approximation to the nearest four decimal places. Similarly, the compound trapezoidal rule will provide an approximation to the nearest two decimal places. However, this approximation will require an additional $\left(\frac{n}{2}\right)$ function evaluation. The number of function evaluations necessary for generating a compound rule using an m-point basic rule for an n-subinterval interval is denoted as $(m \times n)$. Conversely, the quantity of function evaluations diminishes to $n + 1$ when the nodes of the fundamental rule encompass the endpoints of the integration interval. The compound trapezoidal rule is a rule that involves two points with nodes at both ends. For example, if the interval is divided into n-subintervals, then there are $(n + 1)$ function evaluations instead of $2n$.

An indispensable tool in many areas of computer science, composite rules of integration offer a robust and versatile foundation for numerical integration. We can confidently use them to approximatively solve definite integrals if we grasp their construction, error analysis, and applications.

In summary, composite numerical integration rules provide a powerful and flexible approach to approximate definite integrals. By breaking the integration interval into smaller subintervals, these

rules can adapt to the behavior of the function, improve efficiency, and maintain numerical stability. The choice of the specific composite rule and the number of subintervals depends on the characteristics of the integrand and the desired level of accuracy. Error analysis is essential for understanding convergence behavior and ensuring reliable numerical results.

CONCLUSION

In conclusion, numerical integration plays an indispensable role in modern computation, particularly when exact solutions are not feasible. The various methods discussed, from basic quadrature formulas to Gaussian quadrature, each offer unique advantages depending on the nature of the integral being approximated. Newton-Cotes formulas provide a simple and intuitive approach, while Gaussian quadrature stands out for its efficiency and higher degree of accuracy. The inclusion of error analysis is crucial for evaluating the reliability of these methods in practical applications. By exploring the strengths and limitations of different techniques, this article provides valuable insights into selecting appropriate methods for numerical computation, contributing to the broader understanding and development of numerical analysis.

REFERENCES

1. Abd. E.L. 1976. A regularization technique for a class of singular integral problems. 1976.
2. Acharya, B.P. and Das, R.N. 1980. Numerical evaluation of singular integrals of complex valued functions. *J. Math. Sci.*, **15**: 40–44.
3. Acharya, B.P. and Das, R.N. 1981. Numerical determination of cauchy principal value integrals. *Computing,* **27**(4): 373–378.
4. Acharya, M., Acharya, B.P. and Sahoo, S.B. 2013. Numerical computation of cauchy principal value integrals using complex variable techniques. *Applied Mathematical Sciences*, **7**(28): 1367–1373.
5. Ang, W.T. 2007. A beginner's course in boundary element methods. Universal-Publishers.
6. Ang, W.T. 2008. A hypersingular boundary integral formulation for heat conduction across a curved imperfect interface. *Communications in Numerical Methods in Engineering,* **24**(10): 841–851.
7. Ang, W.T. 2014. Hypersingular integral equations in fracture analysis. Elsevier.
8. Atkinson, K. 1991. An introduction to numerical analysis. John wiley & Sons.
9. Chawla, M.M. and Jayarajan, N. 1975. Quadrature formulas for cauchy principal value integrals. *Computing,* **15**(4): 347–355.
10. Conway, J.B. 2012. Functions of one complex variable II, volume 159. Springer Science & Business Media.
11. Criscuolo, G. 1997. A new algorithm for cauchy principal value and hadamard finite-part integrals. *Journal of Computational and Applied Mathematics,* **78**(2): 255–275.
12. Diethelm, K. 1995. Gaussian quadrature formulae of the third kind for cauchy principal value integrals: basic properties and error estimates. *Journal of*

Computational and Applied Mathematics, **65**(1-3): 97–114.

13. Diethelm, K. 1999. A method for the practical evaluation of the hilbert transform on the real line. *Journal of Computational and Applied Mathematics,* **112**(1-2): 45–53.
14. Elliott, D. and Paget, D.F. 1979. Gauss type quadrature rules for cauchy principal value integrals. *Mathematics of Computation,* **33**(145): 301–309.
15. Gradimir V Milovanovi´c, Dobrilo To˘si´c, and Miloljub Albijani´c. 2012. Numerical integration of analytic functions. *In AIP Conference Proceedings,* volume 1479, pages 1046–1049. American Institute of Physics.
16. Lether, F.G. 1971. Error bounds for fully symmetric cubature rules. *SIAM Journal on Numerical Analysis,* **8**(1): 49–60.
17. Li, H. and Chen, K. 2007. Gauss quadrature rules for cauchy principal value integrals with wavelets. *Applied Mathematics and Computation,* **186**(1): 357–364.
18. Liu, G. and Xiang, S. 2023. An efficient quadrature rule for weakly and strongly singular integrals. *Applied Mathematics and Computation,* **447**: 127901.
19. Milovanovic, G.V., Acharya, B.P. and Pattanaik, T.N. 1984. Some interpolatory rules for the approximative evaluation of complex cauchy principal value integrals, review of research. *University ofNovisad, Mathematics Series,* **14**(2): 1984.
20. Okecha, G.E. 1987. Quadrature formulae for cauchy principal value integrals of oscillatory kind. *Mathematics of Computation,* **49**(179): 259–268.
21. Olver, S. 2007. Moment-free numerical approximation of highly oscillatory integrals with stationary points. *European Journal of Applied Mathematics,* **18**(4): 435–447.
22. Orsi, A.P. 1990. Spline approximation for cauchy principal value integrals. *Journal of Computational and Applied Mathematics,* **30**(2): 191–201.
23. Piessens, R. 1970. Numerical evaluation of cauchy principal values of integrals. *BIT Numerical Mathematics,* **10**: 476–480.
24. Wang, H. and Xiang, S. 2010. On the evaluation of cauchy principal value integrals of oscillatory functions. *Journal of Computational and Applied Mathematics,* **234**(1): 95–100.