International Journal of Analysis and Applications

A New Family of Efficient Open-Type Quadrature for the Approximation of Riemann-Stieltjes Integrals Using Derivatives

Khuda Dad[1], Muhammad Mujtaba Shaikh[2,*], Kashif Memon[1], Abdul Wasim Shaikh[1]

[1] *Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Pakistan.*
[2] *Department of Basic Sciences and Related Studies, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan.*
**Corresponding author:* mujtaba.shaikh@faculty.muet.edu.pk

ABSTRACT. Most of the difficulties in control theory and probability distributions are described in terms of the Riemann-Stieltjes (RS) integral rather than the standard Riemann integral (RI). Numerical approximations for the approximation of the RS integral are required due to the nonlinearity of the integrand and the complexity of the analytical process. The numerical techniques, besides convergence features, should also be computationally effective and time-efficient. In this study, some time-efficient and cost effective numerical approaches for approximating the RS integral are proposed. The proposed approximations are based on Newton-Cotes' standard open-type schemes. We derive derivative-based open Newton-Cotes quadrature schemes in both basic and composite forms, as well as the error terms for the Riemann-Stieltjes integral's numerical evaluation. For the suggested schemes, theorems associated with the degree of precision and order of accuracy are studied with proofs. For all suggested and current rules on the test integrals, the absolute error distributions, computational costs, execution times, and computational orders of accuracy have been calculated. To demonstrate the efficacy of the proposed approaches, a numerical verification method will be used. MATLAB R2022a software was used to achieve the results. The proposed method's quick convergence and high efficacy over the current methods have been demonstrated by the results and theoretical properties.

1. Introduction

When analytical methods fail or become extremely difficult to apply to complex problems, mathematicians often search for numerical solutions. However, numerical analysis has developed into an accurate tool for effective and convergent simulations and approximations for

Received Dec. 11, 2025

2020 Mathematics Subject Classification. 65D30, 65D32.

Key words and phrases. quadrature; Riemann-Stieltjes integral; derivative-based scheme.

https://doi.org/10.28924/2291-8639-24-2026-41

ISSN: 2291-8639

a wide range of problems, regardless of their complexity. In numerical computations, the accuracy, efficiency, and convergence of numerical solutions are essential. [1], [3], [5], [8].

The area enclosed by a completely or sectionally continuous curve $f(x)$ and the $X-axis$ in a finite range is the main goal of numerical integration. This idea has been clearly stated in mathematics in terms of a definite integral, which is defined as in (1.1).

$$I(f)=\int_{\upsilon}^{\varphi} f(x)dx \tag{1.1}$$

For functions $f(x) = e^{x^3}$ (or) $cos\, x^3$, the integral in (1.1) has no analytical solution; in these situations, the solution of a definite integral can be found numerically. Quadrature is a method for calculating the value of a definite integral of a function of one variable [1].The well-known Riemann integral (RI), which deals with the geometric concept of area and gives a continuous, non-negative function on the interval $[\upsilon,\varphi]$, is also defined by the integral in (1.1). Thus, a partition of $[\upsilon,\varphi]$ generates sub-rectangles for the curve. The areas of these rectangles are summed together using upper and lower estimations of the area under the curve. The RI, or area under the curve, is often the limit of these estimations as the partition width approaches zero [2].

Whereas the generalized form of the RI is the Riemann-Stieltjes (RS) integral. Assume that two bounded functions defined in the interval $[\upsilon,\varphi]$ exist: $f(x)$ and $\alpha(x)$, with a monotonic increase in $\alpha(x)$. In this context, the definition of the RS integral is:

$$RS(f{:}\,\alpha)\approx\int_{\upsilon}^{\varphi} f(x)d\alpha(x) \tag{1.2}$$

Where the integrator is represented by $\alpha(x)$ and the integrand by $f(x)$ [3].

Recently, there has been a lot of interest in the numerical quadrature rules due to their higher accuracy. When integration cannot be achieved or the function is only known for a limited quantity of data, the Newton-Cotes formulae—closed, open, and semi-open rules—have been studied in the literature as the most fundamental approaches [4], [7], [9], [11–12], [13–15].

The following general form [4] represents a closed Newton-Cotes numerical integration formula for approximating a specified integral of $f(x)$ from $[\upsilon,\varphi]$:

$$\int_{\upsilon}^{\varphi} f(x)dx\approx\sum_{i=0}^{n}\omega_i f(x_i) \tag{1.3}$$

Where the $n+1$ weights and nodes are represented by the ω_i and x_i, respectively. Within the interval [υ, φ], the nodes $\upsilon = x_0, x_1, \dots, x_n = \varphi$ are equipped as equally-spaced points so that $x_i = \upsilon + \mathrm{i\,h}, \mathrm{i} = 0, 1, .. \mathrm{n}$ where $h = \frac{\varphi-\upsilon}{n+2}$, and weights must to be computed using the method of undetermined coefficients. A closed Newton-Cotes numerical integration formula approximates a definite integral of a function $f(x)$ over a finite interval [υ, φ] [4] by incorporating the evaluations of integrand not just within the interval's interior but also the end points of the interval. The initial closed Newton-Cotes quadrature rules are the Trapezoidal, Simpson's 1/3, and Simpson's 3/8 rules, their orders of accuracy in basic forms are 3, 5, and 5, and the degrees of precision are 1, 3, and 3, respectively [5].

The quadrature rules with equally-spaced points of the interval of integration which do not use the function evaluations at the interval's ends are termed as open [4]. The general form is:

$$\int_{\upsilon}^{\varphi} f(x)dx \int_{x_{-1}}^{x_{n+1}} f(x)dx \approx \sum_{i=0}^{n} \omega_i f(x_i) \tag{1.4}$$

Where ω_i are $n+1$ weights within the interval (υ, φ) with $x_i = x_0 + (i+1)h;\ i = 0, 1, 2, \dots, n, and\ h = \frac{\varphi-\upsilon}{n+2}, and\ x_0, x_1, \dots, x_n$ represent unique $n+1$ interior integration points. The most basic form of Newton and Cotes' open-type quadrature is the midpoint rule. In the midpoint formula, the local error order is 3. The degree of exactness for polynomials is just one. It decreases accuracy by one order of accuracy when in composite form [8].

There exist many quadrature schemes for the Riemann and RS integral in literature which are derivative-free as well as derivative-based. The existing schemes for the approximation of the Riemann integral include extensive improvement works on closed, open as well as semi-open closed Newton-Cotes rules. However, when it comes to the RS integral, only some numerical work is done on the closed Newton-Cotes rules. There is quite negligible work on the numerical schemes of open and semi-open Newton-Cotes rules for the RS integral. This is the primary reason for carrying out the current investigation.

A novel derivative-based family of closed Newton-Cotes quadrature rules with error terms was introduced by Zhao and Li [6] in 2013. This family of quadrature rules improved two orders of accuracy compared to the conventional Newton-Cotes quadrature formula. The performance of this family was also discussed in terms of computational cost using three examples.

Shaikh, Chandio and Soomro [7] presented a novel four-point closed quadrature rule in 2016. The mid-point derivative-based Simpson's 3/8 rule was developed by Zhao and Li in [6], and it was a helpful modification. This method decreased Zhao and Li's rule's [6] computational cost without reducing accuracy by using the second-order derivative for the fourth-order derivative at mid-point in each integration strip. In addition, this rule's absolute errors in the three numerical experiments were less than those of Zhao and Li's rule [6] and the original based on the midpoint derivative.

In 2015, Zhao, Zhang and Ye [9] discussed the global error terms for the composite form of the trapezoid RS integral and proposed the trapezoid rule for RS integrals.

In 2014, Zhao and Zhang [10] discussed a derivative-based trapezoid technique to split the RS integral into two derivative values at the endpoints. The RS integral was discussed, and two precisions were increased compared to the trapezoid scheme. The precision of the method was three.

In 2021, Zhao and Zhang [11] developed Simpson's scheme for the RS integral. The method had a lower precision of order two than Simpson's method for the Riemann integral. The method was third order accurate.

In 2023, Zhao and Zhang [12] presented a new derivative-based trapezoid rule for a special kind of RS integral with error terms, which used two derivative values at the endpoints. This quadrature rule improved two orders of accuracy over the trapezoid formula. The performance of this family was also discussed in terms of computational cost using some numerical examples.

In 2014, Zhao and Zhang [13] offered a novel trapezoid rule for RS integrals based on derivatives with error terms, which used derivative values at the midpoint. This quadrature rules improved two orders of accuracy over the trapezoid rule for the RS.

A new quadrature scheme of derivative-free Simpson 1/3-type for the RS integral approximation using error terms in both basic and composite forms was discussed using the concept of precision by Memon et al. in 2020 [14]. Furthermore, they discussed with numerical problems and compared their proposed method with others RS integral in terms of costs and time efficiency. Numerical results show that, for each test problem, the proposed method is significantly more efficient than existing schemes.

A novel Simpson's 1/3 technique for the RS integral based on the harmonic mean derivative was introduced in [15]. The proposed scheme's basic and composite forms with local and global error terms had been developed. With $g(t) = t$ in the derived formula for RS integral, the reduction to the corresponding Riemann integral scheme was attained. There has been discussion of experimental work in the paper to compare the new proposed scheme results to those of other schemes using MATLAB. The new suggested scheme's calculations were made for the order of accuracy, computational cost, and average CPU time (in seconds). Lastly, computational study show that the proposed technique improved existing methods.

In 2006, Dehghan, Masjed-Jamei and Eslahchi [16] improved the open type of Newton-Cotes integration rules, in which the integral was limited to take two additional variables, a and h, and numerically improved the precision up to degree $n + 2$. Finally, authors observed that the approximate numerical tests illustrated the numerical superiority of the new method over the usual Newton-Cotes open integration rules.

In 2005, Dehghan, Masjed-Jamei and Eslahchi [17] improved the semi-open type of Newton-Cotes integration rules, in which the integral was limited to take two additional variables, a and h, and numerically improved the precision up to degree $n + 2$. Finally, authors observed that the approximate numerical tests illustrated the numerical superiority of the new method over the usual semi-open Newton-Cotes integration rules.

In 2004, Dehghan , Masjed-Jamei and Eslahchi [18] improved the closed type of Newton-Cotes integration rules, in which the integral was limited to take two additional variables, a and h, and numerically improved the precision up to degree $n+2$. Finally, authors observed that the approximate numerical tests illustrated the numerical superiority of the new method over the usual Newton-Cotes closed integration rules.

In 2005, Babolian , Masjed-Jamei and Eslahchi [19] improved the Gauss–Legendre quadrature rules, in which the integral was limited to take two additional variables, a and b, and numerically improved the precision. New n-point quadrature rules are produced when the Gauss–Legendre system of non-linear equations changed into an expanded system.

A three-point numerical technique adding the derivative in each strip at the geometric mean of the interval of integration's end points is theoretically analyze in order to approximate a RS integral was the main objective of this study [20]. Memon et al. in 2025 [21] used a four-point Simpson's rule with a midpoint derivative to develop a novel numerical quadrature scheme for the RS integral. Theorems for local and global error terms were established, and the scheme was developed in both basic and composite forms. In 2023, Marjulisa and Imran [22] discussed the derivation of the double midpoint rule for approximating the RS integral. They derived the double midpoint rule by approximating certain monomial functions, which then yielded the values of the new method's coefficients. In 2022, Mahesar et al. [23] developed several new semi-open-type rules with derivatives, with a focus on the cost and time efficiency of their computation. These rules included evaluations of the function and its first derivative at data points that are equally spaced. In 2023, Mahesar et al. [24], for the computation of definite integrals, developed a new family of open Newton-Cotes quadrature methods based on the centroidal mean were developed. The modified methods' error terms were derived from the precision concept.

Over the years, many derivative-free and derivative-based quadrature rules have been proposed in the literature for the Riemann integral as well as the Riemann-Stieltjes integral. But the schemes were closed Newton-Cotes types; the open rules were never suggested for the Riemann-Stieltjes integral. There are some recent derivative-based open rules with increased efficiency over the derivative-free open rules for only the Riemann integral. It will be interesting to see some new schemes with open-type derivatives for the numerical evaluation of the Riemann-Stieltjes integral.

2. Material and methods

2.1 Existing rules for Riemann-Stieltjes integral (RSI)

In [25], authors presented the following trapezoid-type (T) basic-form RSI approximation with error term:

$$\int_{\upsilon}^{\varphi} f dg \approx [N - g(\upsilon)]f(\upsilon) + [g(\varphi) - N]f(\varphi)] - \frac{(\varphi-\upsilon)^3}{12} f''(\xi)g'(\sigma) \quad (2.1)$$

Where $N = \frac{1}{\varphi-\upsilon}\int_{\upsilon}^{\varphi} g(t)dt \; and \; \xi, \sigma \in (\upsilon\,, \varphi)$ The degree of precision of T is one. The corresponding composite scheme, referred as CT through [16] is stated as:

$$\mathrm{RS(f{:}\,g)} \approx CT = \begin{bmatrix} \frac{n}{\varphi-\upsilon} N_1(\upsilon\,, x_1) \\ -g(\upsilon) \end{bmatrix} f(\upsilon) + \frac{n}{\varphi-\upsilon}\sum_{l=1}^{n-1}[N_1(x_{l-1}\,, x_{l+1}) - N_1(x_{l-1}\,, x_l)]\, f(x_l)$$

$$+\left[g(\varphi) - \frac{n}{\varphi-\upsilon}\, N_1\left(x_{n-1}\,, x_{\varphi}\right)\right] f(\varphi) \text{ where } N_1(\upsilon\,, \varphi) = \int_{\upsilon}^{\varphi} g(t)dt \quad (2.2)$$

The error term for composite RSI trapezoid rule (2.2) is: $-\frac{(\varphi-\upsilon)^3}{12n^2} f''(\mu)g'(\eta)$ Where μ and η are in $(\upsilon\,, \varphi)$.

In [6], authors presented a modified trapezoid-type basic-form RSI approximation, denoted as ZT, and defined as:

$$RS(f{:}\,g) \approx ZT = T + \left(N_2(\upsilon\,, \varphi) - \frac{\varphi-\upsilon}{2} N_1(\upsilon\,, \varphi)\right) f''(c) \quad (2.3)$$

where $c = \frac{(-2\varphi^2 + \upsilon^2 - \upsilon\varphi)N_1(\upsilon\,,\varphi) + 6\varphi N_2(\upsilon\,,\varphi) - 6N_3(\upsilon\,,\varphi)}{6N_2(\upsilon\,,\varphi) - 3(\varphi-\upsilon)N_1(\upsilon\,,\varphi)}$,

$N_2(\upsilon\,, \varphi) = \iint_{\upsilon\,\upsilon}^{\varphi\; t} g(x)dxdt$ and $N_3(\upsilon\,, \varphi) = \int_{\upsilon}^{\varphi}\int_{\upsilon}^{t}\int_{\upsilon}^{y} g(x)dxdydt$

This approach has a precision degree three. The error term is:

$$R_{ZCT}[f] = \begin{pmatrix} \frac{\upsilon^3 + \upsilon\varphi^2 + \upsilon^2\varphi - 3\varphi^3 + 6(\varphi-\upsilon)c^2}{24} N_1(\upsilon\,, \varphi) \\ + \frac{\varphi^2 - \mathrm{c}^2}{2} N_2(\upsilon\,, \varphi) - \varphi N_3(\upsilon\,, \varphi) + N_4(\upsilon\,, \varphi) \end{pmatrix} f^{(4)}(\xi)g'(\eta) \quad (2.4)$$

Where ξ and $\eta \in (\upsilon\,, \varphi) \; and \; N_4(\upsilon\,, \varphi) = \int_{\upsilon}^{\varphi}\int_{\upsilon}^{t}\int_{\upsilon}^{z}\int_{\upsilon}^{y} g(x)dxdydzdt$. The composite form is:

$$\mathrm{RS(f{:}\,g)} \approx \mathrm{CZCT} = \mathrm{CT} + \sum_{l=1}^{n}\left[N_2(x_{l-1}\,, x_l) - \frac{h}{2}N_1(x_{l-1}\,, x_l)\right] f''(c_l) \quad (2.5)$$

$$c_l = \frac{\left(-2x_l^2 + x_{l-1}^2 - x_{l-1}x_l\right)N_1(x_{l-1}\,, x_l) + 6\varphi N_2(x_{l-1}\,, x_l) - 6N_3(x_{l-1}\,, x_l)}{6N_2(x_{l-1}\,, x_l)N_2(x_{l-1}\,, x_l) - \frac{3(x_l - x_{l-1})}{n} N_1(x_{l-1}\,, x_l)}$$

The error term is:

$$R_{CZCT}[f] = n\begin{pmatrix} \frac{\upsilon^3 + \upsilon\varphi^2 + \upsilon^2\varphi - 3\varphi^3 + 6(\varphi-\upsilon)c_l^2}{24} N_1(\chi\,, \varphi) \\ + \frac{\varphi^2 - \mathrm{c}_\mathrm{l}^2}{2} N_2(\upsilon\,, \varphi) - \varphi, N_3(\upsilon\,, \varphi) + N_4(\upsilon\,, \varphi) \end{pmatrix} f^{(4)}(\mu)g'(\eta) \quad (2.6)$$

Where μ and $\eta \in (\upsilon\,,\ \varphi)$

In [7], authors presented a correction to the scheme ZT, denoted as MZT. This trapezoid-type basic-form RSI approximation is defined as:

$$RS(f\colon g)\ \approx\ MZT\ =\ T + \left(N_2(\upsilon\,,\varphi) - \tfrac{\varphi-\upsilon}{2} N_1(\upsilon\,,\varphi)\right) f''(c) \tag{2.7}$$

where $c = \dfrac{(-2\varphi^2 + \upsilon^2 + \upsilon\varphi)N_1(\upsilon\,,\varphi) + 6\varphi N_2(\upsilon\,,\varphi) - 6N_3(\upsilon\,,\varphi)}{6N_2(\upsilon\,,\varphi) - 3(\varphi-\upsilon)N_1(\upsilon\,,\varphi)}$. The error term is:

$$R_{MZCT}[f] = \begin{pmatrix} \dfrac{\upsilon^3 + \upsilon\varphi^2 + \upsilon^2\varphi - 3\varphi^3 + 6(\varphi-\upsilon)c^2}{24} N_1(\upsilon\,,\varphi) \\ + \dfrac{\varphi^2 - \mathrm{c}^2}{2} N_2(\upsilon\,,\varphi) - \varphi N_3(\upsilon\,,\varphi) + N_4(\upsilon\,,\varphi) \end{pmatrix} f^{(4)}(\xi) \tag{2.8}$$

Where $\xi\ \in\ (\upsilon\,,\varphi)$. The composite form is:

$$\mathrm{RS(f\colon g)} \approx\ \mathrm{MZCT}\ =\ \mathrm{CT} + \sum_{l=1}^{n}\left[N_2(x_{l-1}\,,x_l) - \frac{h}{2}N_1(x_{l-1}\,,x_l)\right] f''(c_l) \tag{2.9}$$

$$c_l = \frac{\left(-2x_l^2 + x_{l-1}^2 + x_{l-1}x_l\right)N_1(x_{l-1}\,,x_l) + 6\varphi N_2(x_{l-1}\,,x_l) - 6N_3(x_{l-1}\,,x_l)}{6N_2(x_{l-1}\,,x_l) - \frac{3(x_l - x_{l-1})}{n}N_1(x_{l-1}\,,x_l)}$$

The error term is:

$$R_{CMZCT}[f] = n\begin{pmatrix} \dfrac{\upsilon^3 + \upsilon\varphi^2 + \upsilon^2\varphi - 3\varphi^3 + 6(\varphi-\upsilon)c_l^2}{24} N_1(\upsilon\,,\varphi) \\ + \dfrac{\varphi^2 - c_l^2}{2} N_2(\upsilon\,,\varphi) - \varphi N_3(\upsilon\,,\varphi) + N_4(\upsilon\,,\varphi) \end{pmatrix} f^{(4)}(\mu)g'(\eta) \tag{2.10}$$

In [14], authors presented the following derivative based Simpson's type RSI approximation using harmonic mean at derivative is:

$$\mathrm{RS(f\colon g)} \approx \mathrm{HM} = \begin{bmatrix} \dfrac{4}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \\ -\dfrac{1}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) - g(\upsilon) \end{bmatrix} f(\chi) + \begin{bmatrix} \dfrac{4}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) \\ -\dfrac{8}{(\varphi-\upsilon)^2} N_1(\upsilon\,,\varphi) \end{bmatrix} f\left(\frac{\upsilon+\varphi}{2}\right)$$

$$+\begin{bmatrix} g(\varphi) \\ -\dfrac{3}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) \\ +\dfrac{4}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \end{bmatrix} f(\varphi) + \begin{bmatrix} \dfrac{-(\varphi-\upsilon)^2(3\upsilon+5\varphi)}{96} N_1(\upsilon\,,\varphi) \\ +\dfrac{17\varphi^2 - 10\upsilon\varphi - 7\upsilon^2}{48} N_2(\upsilon\,,\varphi) - \\ \varphi N_3(\upsilon\,,\varphi) + N_4(\upsilon\,,\varphi) \end{bmatrix} f^{(4)}\left(\frac{2\upsilon\varphi}{\upsilon+\varphi}\right) \tag{2.11}$$

In [24], authors presented the following derivative based Simpson's type RSI approximation using Centroidal mean at derivative is:

$$\mathrm{RS(f\colon g)} \approx \mathrm{CM} = \begin{bmatrix} \dfrac{4}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \\ -\dfrac{1}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) - g(\upsilon) \end{bmatrix} f(\upsilon) + \begin{bmatrix} \dfrac{4}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) \\ -\dfrac{8}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \end{bmatrix} f\left(\frac{\upsilon+\varphi}{2}\right)$$

$$+\left[g(\varphi) - \frac{3}{\varphi-\upsilon}\ N_1(\upsilon\,,\varphi) +\ \frac{4}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi)\right] f(\varphi)$$

$$+\begin{bmatrix} \dfrac{-(\varphi-\upsilon)^2(3\upsilon+5\varphi)}{96}N_1(\upsilon\,,\varphi) \\ +\dfrac{17\varphi^2-10\upsilon\varphi-7\upsilon^2}{48}N_2(\upsilon\,,\varphi)-\varphi N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{bmatrix} f^{(4)}\left(\frac{2(\upsilon^2+\upsilon\varphi+\varphi^2)}{3(\upsilon+\varphi)}\right) \tag{2.12}$$

In [24], authors presented the following derivative based Simpson's type RSI approximation using Heronian mean at derivative is:

$$\mathrm{RS(f{:}\,g)} \approx \mathrm{HeM} = \begin{bmatrix} \dfrac{4}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) \\ -\dfrac{1}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi)-g(\upsilon)\end{bmatrix} f(\upsilon)+\begin{bmatrix} \dfrac{4}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi) \\ -\dfrac{8}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)\end{bmatrix} f\left(\frac{\upsilon+\varphi}{2}\right)$$

$$+\left[g(\varphi)-\frac{3}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi)+\frac{4}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)\right]f(\varphi)$$

$$+\begin{bmatrix} \dfrac{-(\varphi-\upsilon)^2(3\upsilon+5\varphi)}{96}N_1(\upsilon\,,\varphi) \\ +\dfrac{17\varphi^2-10\upsilon\varphi-7\upsilon^2}{48}N_2(\upsilon\,,\varphi)-\varphi N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{bmatrix} f^{(4)}\left(\frac{\upsilon+\sqrt{\upsilon\varphi}+\varphi}{3}\right) \tag{2.13}$$

In [20], authors presented the following derivative based Simpson's type RSI approximation using geometric mean at derivative is:

$$\mathrm{RS(f{:}\,g)} \approx \mathrm{GM} = \begin{bmatrix} \dfrac{4}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) \\ -\dfrac{1}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi)-g(\upsilon)\end{bmatrix} f(\upsilon)+\begin{bmatrix} \dfrac{4}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi) \\ -\dfrac{8}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)\end{bmatrix} f\left(\frac{\upsilon+\varphi}{2}\right)$$

$$+\begin{bmatrix} g(\varphi)-\dfrac{3}{\varphi-\upsilon}\,N_1(\upsilon\,,\varphi) \\ +\dfrac{4}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)\end{bmatrix} f(\varphi)+\begin{bmatrix} \dfrac{-(\varphi-\upsilon)^2(3\upsilon+5\varphi)}{96}N_1(\upsilon\,,\varphi) \\ +\dfrac{17\varphi^2-10\upsilon\varphi-7\upsilon^2}{48}N_2(\upsilon\,,\varphi) \\ -\varphi N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{bmatrix} f^{(4)}\left(\sqrt{\upsilon\varphi}\right) \tag{2.14}$$

2.2. Proposed derivative based open rules for RS Integral

In this research two conventional derivative-based open Newton-Cotes (ONC) schemes are extended for the RS integral, one after one, and finally the derivations and theorems concerning the new schemes are also proven. The numerical results and discussion are presented at the end of this paper.

2.2.1 Open Newton-Cotes rules with derivative at all interior nodes [4]

The starting three members of the first family of derivative-based ONC rules presented in [4] are improvements of the derivative-based ONC rules: midpoint (M), open trapezoid(OT) and Milne's (ML) rules. The improvement was achieved by using the derivative at all interior points.

Recalling that the derivative-based midpoint ONC method $(DBM1)$ with local error term for RI is:

$$I(f) \approx DBM1 + R_{DBM1} = (\varphi-\upsilon)f\left(\frac{\upsilon+\varphi}{2}\right)+\frac{(\varphi-\upsilon)^3}{24}f^{(2)}(\xi),\ \xi\ \in\ (a,\varphi) \tag{2.15}$$

This is the same as mid-point rule.

2.2.1.1 Proposed Scheme-1 (PS1) for RSI

Recalling that the derivative-based open trapezoidal ONC method ($DBOT1$) with local error term for RI is:

$$I(f) \approx DBOT1 + R_{DBOT1} = OT - \frac{3}{12}(\varphi - \upsilon)^2 \begin{bmatrix} f'\left(\frac{2\upsilon + \varphi}{3}\right) \\ -f'\left(\frac{\upsilon + 2\varphi}{3}\right) \end{bmatrix} + \frac{7(\varphi - \upsilon)^5}{19440} f^{(4)}(\xi) \tag{2.16}$$

Now the extension of scheme (2.16) to RSI is explained in the forthcoming sections, basic from in Theorem 2.1. The result connected to the derivation of PS1's local error is explained in Theorem 2.2. In Theorem 2.3, we verify that, in the trivial case of integrator $g\,(t) = t$, the RSI approximation through PS1 with error term reduces to the comparable conclusion for the RI. Lastly, the global form of the error term and the extension to composite form of the suggested PS1 approximation of RSI are explained with proofs in Theorems 2.4–2.5.

***Theorem* 2.1** *Assuming that $f(t)$ and $g(t)$ are continuous on (υ, φ) and that $g\,(t)$ increases there, the PS1 for RSI approximation is:*

$$RS(f\colon g) \approx PS1 = \begin{pmatrix} 5g(\varphi) + 4g(\upsilon) - \frac{36}{\varphi - \upsilon} N_1(\upsilon\,, \varphi) \\ + \frac{162}{(\varphi - \upsilon)^2} N_2(\upsilon\,, \varphi) - \frac{324}{(\varphi - \upsilon)^3} N_3(\upsilon\,, \varphi) \end{pmatrix} f\left(\frac{2\upsilon + \varphi}{3}\right)$$
$$+ \begin{pmatrix} -4g(\varphi) - 5g(\upsilon) \\ + \frac{36}{\varphi - \upsilon} N_1(\upsilon\,, \varphi) \\ - \frac{162}{(\varphi - \upsilon)^2} N_2(\upsilon\,, \varphi) \\ + \frac{324}{(\varphi - \upsilon)^3} N_3(\upsilon\,, \varphi) \end{pmatrix} f\left(\frac{\upsilon + 2\varphi}{3}\right) + \begin{pmatrix} \frac{2(\varphi - \upsilon)}{3} g(\varphi) + \frac{4(\varphi - \upsilon)}{3} g(\upsilon) \\ -5N_1(\upsilon\,, \varphi) + \frac{24}{\varphi - \upsilon} N_2(\upsilon\,, \varphi) \\ - \frac{54}{(\varphi - \upsilon)^2} N_3(\chi\,, \varphi) \end{pmatrix} f'\left(\frac{2\upsilon + \varphi}{3}\right)$$
$$+ \left(\frac{4(\varphi - \upsilon)}{3} g(\varphi) + \frac{2(\varphi - \upsilon)}{3} g(\upsilon) - 8N_1(\upsilon\,, \varphi) \right) f'\left(\frac{\upsilon + 2\varphi}{3}\right)$$
$$+ \frac{30}{\varphi - \upsilon} N_2(\upsilon\,, \varphi) - \frac{54}{(\varphi - \upsilon)^2} N_3(\upsilon\,, \varphi) \tag{2.17}$$

***Proof of theorem* 2.1**

Looking for the RSI we find, a_0 ,$b_{0\,,}$ c_0 and d_0 such that:

$$\int_{\upsilon}^{\varphi} f(t)dg \approx a_0 f\left(\frac{2\upsilon + \varphi}{3}\right) + b_0 f\left(\frac{\upsilon + 2\varphi}{3}\right) + c_0 f'\left(\frac{2\upsilon + \varphi}{3}\right) + d_0 f'\left(\frac{\upsilon + 2\varphi}{3}\right) \tag{2.18}$$

is exact for $f(t) = 1,\ t\,, t^2 and\ t^3$. Using these in (2.18), we get two equations:

$$\int_{\upsilon}^{\varphi} 1dg \approx a_0 + b_0$$

$$\int_{\upsilon}^{\varphi} tdg \approx a_0\left(\frac{2\upsilon+\varphi}{3}\right) + b_0\left(\frac{\upsilon+2\varphi}{3}\right) + c_0 + d_0$$

$$\int_{\upsilon}^{\varphi} t^2dg \approx a_0\left(\frac{2\upsilon+\varphi}{3}\right)^2 + b_0\left(\frac{\upsilon+2\varphi}{3}\right)^2 + 2c_0\left(\frac{2\upsilon+\varphi}{3}\right) + 2d_0\left(\frac{\upsilon+2\varphi}{3}\right)$$

$$\int_{\upsilon}^{\varphi} t^3dg \approx a_0\left(\frac{2\upsilon+\varphi}{3}\right)^3 + b_0\left(\frac{\upsilon+2\varphi}{3}\right)^3 + 3c_0\left(\frac{2\upsilon+\varphi}{3}\right)^2 + 2d_0\left(\frac{\upsilon+2\varphi}{3}\right)^2$$

We get the following by applying integration by parts of the RSI, as in [10]:

$$g(\varphi) - g(\upsilon) = a_0 + b_0 \tag{2.19}$$

$$\varphi g(\varphi) - ag(\upsilon) - N_1(\upsilon\,,\varphi) = a_0\left(\frac{2\upsilon+\varphi}{3}\right) + b_0\left(\frac{\upsilon+2\varphi}{3}\right) + c_0 + d_0 \tag{2.20}$$

$$\begin{aligned}\varphi^2 g(\varphi) - \upsilon^2 g(\chi) - 2\varphi N_1(\upsilon\,,\varphi) + 2N_2(\upsilon\,,\varphi) &= a_0\left(\frac{2\upsilon+\varphi}{3}\right)^2 + b_0\left(\frac{\upsilon+2\varphi}{3}\right)^2 \\ &\quad +2c_0\left(\frac{2\upsilon+\varphi}{3}\right) + 2d_0\left(\frac{\upsilon+2\varphi}{3}\right)\end{aligned} \tag{2.21}$$

$$\begin{aligned}&\varphi^3 g(\varphi) - \upsilon^3 g(\upsilon) - 3\varphi^2 N_1(\upsilon\,,\varphi) + 6\varphi N_2(\upsilon\,,\varphi) - 6N_3(\upsilon\,,\varphi) \\ &= a_0\left(\tfrac{2\upsilon+\varphi}{3}\right)^3 + b\,_0\left(\tfrac{\upsilon+2\varphi}{3}\right)^3 + 3c_0\left(\tfrac{2\upsilon+\varphi}{3}\right)^2 + 2d_0\left(\tfrac{\upsilon+2\varphi}{3}\right)^2\end{aligned} \tag{2.22}$$

The co-efficient matrix of the system of linear equation (2.19) to (2.22) can be expressed as M and M_R is reduced echelon from

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ \left(\frac{2\upsilon+\varphi}{3}\right) & \left(\frac{\upsilon+2\varphi}{3}\right) & 1 & 1 \\ \left(\frac{2\upsilon+\varphi}{3}\right)^2 & \left(\frac{\upsilon+2\varphi}{3}\right)^2 & 2\left(\frac{2\upsilon+\varphi}{3}\right) & 2\left(\frac{\upsilon+2\varphi}{3}\right) \\ \left(\frac{2\upsilon+\varphi}{3}\right)^3 & \left(\frac{\upsilon+2\varphi}{3}\right)^3 & 3\left(\frac{2\upsilon+\varphi}{3}\right)^2 & 3\left(\frac{\upsilon+2\varphi}{3}\right)^2 \end{bmatrix}$$

$$= M \overset{R}{\approx} M_R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As in M_R, the first three rows are nonzero rows, therefore M has three linearly independent rows, and rank $(M) = 4$. To determine the coefficients a_0, b_0 , c_0 and d_0 we solve equations (2.19), (2.20), (2.21) and (2.22) simultaneously, to have

$$a_0 = 5g(\varphi) + 4g(\upsilon) - \frac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) + \frac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) - \frac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)$$

$$b_0 = -4g(\varphi) - 5g(\upsilon) + \tfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) - \tfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) + \tfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)$$

$$c_0 = \frac{2(\varphi-\upsilon)}{3}g(\varphi) + \frac{4(\varphi-\upsilon)}{3}g(\upsilon) - 5N_1(\upsilon\,,\varphi) + \frac{24}{\varphi-\upsilon}N_2(\upsilon\,,\varphi) - \frac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi)$$

$$d_0=\frac{4(\varphi-\upsilon)}{3}g(\varphi)+\frac{2(\varphi-\upsilon)}{3}g(\upsilon)-8N_1(\upsilon\,,\varphi)+\frac{30}{\varphi-\upsilon}N_2(\upsilon\,,\varphi)-\frac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi)$$

Using the first result, and putting the expression of a_0 ,b_0 c_0 and d_0 in equation (2.18), we get final form of PS1:

$$\mathrm{RS(f{:}\ g)}\ \approx\ \mathrm{PS1}\ =\begin{pmatrix} 5g(\varphi)+4g(\upsilon)-\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ +\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)-\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)\end{pmatrix}f\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} -4g(\varphi)-5g(\upsilon) \\ +\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ -\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) \\ +\dfrac{324}{(\varphi-\upsilon)^3}N_1(\upsilon\,,\varphi)\end{pmatrix}f\left(\frac{\upsilon+2\varphi}{3}\right)+\begin{pmatrix} \dfrac{2(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{4(\varphi-\upsilon)}{3}g(\upsilon) \\ -5N_1(\upsilon\,,\varphi)+\dfrac{24}{\varphi-\upsilon}N_2(\upsilon\,,\varphi) \\ -\dfrac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi)\end{pmatrix}f'\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} \dfrac{4(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{2(\varphi-\upsilon)}{3}g(\upsilon)-8N_1(\upsilon\,,\varphi) \\ +\dfrac{30}{\varphi-\upsilon}N_2(\upsilon\,,\varphi)-\dfrac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi)\end{pmatrix}f'\left(\frac{\upsilon+2\varphi}{3}\right)\ \blacksquare$$

***Theorem* 2.2:** *Assuming that $f\ (t)$ and $g\ (t)$ are continuous on (υ, φ) and that $g\ (t)$ increases there, the local error term in PS1 for the RSI approximation is:*

$$R_{PS1}[f]=\begin{bmatrix} \dfrac{(\varphi-\upsilon)^4}{486}\ g(\varphi)-\dfrac{(\varphi-\upsilon)^4}{486}g(\upsilon)-\dfrac{(\varphi-\upsilon)^3}{54}N_1(\upsilon\,,\varphi) \\ +\dfrac{13(\varphi-\upsilon)^2}{108}N_2(\upsilon\,,\varphi)-\dfrac{(\varphi-\upsilon)}{2}N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{bmatrix}f^4(\mu)g'(\eta) \qquad (2.23)$$

where $\xi,\sigma\in\ (\upsilon,\varphi)$.

***Proof of theorem* 2. 2**

Because the precision of scheme (2.16) for RI was 1, then so is for the PS1 for RSI. Using $f(t)=\frac{t^2}{2!}$, which will be the basis for the leading term in the truncation error for the PS1 approximation for the RSI, we state the error term as:

$$R_{PS1}[f]=\frac{1}{4!}\int_{\upsilon}^{\varphi}t^4dg-PS4\left(\frac{t^4}{4!};g;\upsilon,\varphi\right) \qquad (2.24)$$

Using the results from [28], we can write:

$$\frac{1}{4!}\int_{\upsilon}^{\varphi}t^4dg=\frac{1}{24}\left(\varphi^4g(\varphi\ )-\upsilon^4g(\upsilon)\right)-\frac{\varphi^3}{6}N_1(\upsilon\,,\varphi)+\frac{\varphi^2}{2}N_2(\upsilon\,,\varphi)-\varphi N_3(\upsilon\,,\varphi)$$
$$+N_4(\upsilon\,,\varphi)\quad (2.25)$$

From Theorem 2.1, using PS1 approximation for the RSI we can write:

$$PS4\left(\frac{t^4}{4!};g;\upsilon,\varphi\right)=\begin{pmatrix} 5g(\varphi)+4g(\upsilon)-\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ +\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)-\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)\end{pmatrix}\frac{(2\upsilon+\varphi)^4}{4!\,3^4}$$

$$+\begin{pmatrix} -4g(\varphi)-5g(\upsilon)+\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ -\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) \\ +\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)\end{pmatrix}\frac{(\upsilon+2\varphi)^4}{4!\,3^4}+\begin{pmatrix} \dfrac{2(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{4(\varphi-\upsilon)}{3}g(\upsilon) \\ -5N_1(\upsilon\,,\varphi)+\dfrac{24}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ -\dfrac{54}{(\varphi-\upsilon)^2}N_1(\upsilon\,,\varphi)\end{pmatrix}\frac{(2\upsilon+\varphi)^3}{3!\,3^3}$$

$$+\begin{pmatrix} \dfrac{4(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{2(\varphi-\upsilon)}{3}g(\upsilon)-8N_1(\upsilon\,,\varphi)+\dfrac{30}{\varphi-\upsilon}N_2(\upsilon\,,\varphi) \\ -\dfrac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi)\end{pmatrix}\frac{(\upsilon+2\varphi)^3}{3!\,3^3} \qquad (2.26)$$

$$PS1\left(\frac{t^4}{3!};g;\upsilon,\varphi\right)=\left(\frac{-4\upsilon^4+16\upsilon^3\varphi-24\upsilon^2\varphi^2+16\upsilon\varphi^3+77\varphi^4}{1944}\right)g(\varphi)$$

$$+\left(\frac{-77\upsilon^4-16\upsilon^3\varphi+24\upsilon^2\varphi^2-16\upsilon\varphi^3+4\varphi^4}{1944}\right)g(\upsilon)+\left(\frac{\begin{matrix}-36\upsilon^3+108\upsilon^2 b\\ -108\upsilon\varphi^2-288\varphi^3\end{matrix}}{19443846}\right)N_1(\upsilon\,,\varphi)$$

$$+\left(\frac{-13\upsilon^2+41\varphi^2+26\upsilon\varphi}{108}\right)N_2(\upsilon\,,\varphi)-\frac{(\varphi+\upsilon)}{2}N_3(\upsilon\,,\varphi) \qquad (2.27)$$

By putting values of (2.26) and (2.27) in equation (2.25) we get:

$$R_{PS1}[f]=\begin{bmatrix} \dfrac{(\varphi-\upsilon)^4}{486}\;g(\varphi)-\dfrac{(\varphi-\upsilon)^4}{486}g(\upsilon)-\dfrac{(\varphi-\upsilon)^3}{54}N_1(\upsilon\,,\varphi) \\ +\dfrac{13(\varphi-\upsilon)^2}{108}N_2(\upsilon\,,\varphi)-\dfrac{(\varphi-\upsilon)}{2}N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{bmatrix}f^4(\mu)g'(\eta) \qquad \blacksquare$$

Theorem* 2. *3: *With* $g(t)\;=\;t$, *PS*1 *through* (2.17) *and corresponding basic form error through* (2.21) *for the RSI approximation lead to classical DBOT1 approximation* (2.16) *for RI.*

Proof of theorem* 2.*3

Using the results in (2.18) and (2.21), the complete PS1 approximation for the RSI is:

$$PS1(f\colon g)=PS1+R_{PS1}[f]=\begin{pmatrix} 5g(\varphi)+4g(\upsilon)-\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ +\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi)-\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)\end{pmatrix}f\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} -4g(\varphi)-5g(\upsilon) \\ +\dfrac{36}{\varphi-\upsilon}N_1(\upsilon\,,\varphi) \\ -\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon\,,\varphi) \\ +\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon\,,\varphi)\end{pmatrix}f\left(\frac{\upsilon+2\varphi}{3}\right)\;+\begin{pmatrix} \dfrac{2(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{4(\varphi-\upsilon)}{3}g(\upsilon) \\ -5N_1(\upsilon\,,\varphi)\dfrac{24}{\varphi-\upsilon}N_2(\upsilon\,,\varphi) \\ -\dfrac{54}{(\varphi-\upsilon)^{\,2}}N_3(\upsilon\,,\varphi)\end{pmatrix}f'\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} \frac{4(\varphi-\upsilon)}{3}g(\varphi)+\frac{2(\varphi-\upsilon)}{3}g(\upsilon) \\ -8N_1(\upsilon\,,\varphi)+\frac{30}{\varphi-\upsilon}N_2(\upsilon\,,\varphi) \\ -\frac{54}{(\varphi-\upsilon)^2}N_3(\upsilon\,,\varphi) \end{pmatrix} f'\left(\frac{\upsilon+2\varphi}{3}\right)+\begin{bmatrix} \frac{(\varphi-\upsilon)^4}{486}\ g(b) \\ -\frac{(\varphi-\upsilon)^4}{486}g(\upsilon) \\ -\frac{(\varphi-\upsilon)^3}{54}N_1(\upsilon\,,\varphi) \\ +\frac{13(\varphi-\upsilon)^2}{108}N_2(\upsilon\,,\varphi) \\ -\frac{(\varphi-\upsilon)}{2}N_3(\upsilon\,,\varphi) \\ +N_4(\upsilon\,,\varphi) \end{bmatrix} f^4(\xi)g'(\eta) \quad (2.28)$$

It is easy to obtain with reference to [10]:

$\int_{\upsilon}^{\varphi} dt = \varphi-\upsilon$

$N_1(\upsilon\,,\varphi)=\frac{\varphi^2-\upsilon^2}{2}$

$N_2(\upsilon\,,\varphi)=\frac{\varphi^3}{6}-\frac{\upsilon\varphi}{2}+\frac{\upsilon^3}{3}$

$N_3(\upsilon\,,\varphi)=\frac{\varphi^4}{24}-\frac{\upsilon^2\varphi^2}{4}+\frac{\upsilon^3\varphi}{3}-\frac{\upsilon^4}{8}$

$N_4(\upsilon\,,\varphi)=\frac{\varphi^5}{120}-\frac{\upsilon^2\varphi^3}{12}+\frac{\upsilon^3\varphi^2}{3}-\frac{\upsilon^4\varphi}{8}+\frac{\upsilon^5}{30}$

Using $g(t) = t, \varphi - \upsilon = 2h$, and putting these in equation (2.27), we get:

$$\mathrm{PS1(f{:}\,t)} = \left(\frac{\varphi-\upsilon}{2}\right)f\left(\frac{2\upsilon+\varphi}{3}\right)+\left(\frac{\varphi-\upsilon}{2}\right)f\left(\frac{\upsilon+2\varphi}{3}\right) + \left(-\frac{(\varphi-\upsilon)^2}{12}\right)f'\left(\frac{2\upsilon+\varphi}{3}\right) + \left(\frac{(\varphi-\upsilon)^2}{12}\right)f'\left(\frac{\upsilon+2\varphi}{3}\right)+\left[\frac{7(\varphi-\upsilon)^5}{19440}\right] f^4(\xi)$$

$$\mathrm{PS1(f{:}\,t)} = \frac{3h}{2}f\left(\frac{2\upsilon+\varphi}{3}\right)+\frac{3h}{2}f\left(\frac{\upsilon+2\varphi}{3}\right)-\frac{3h^2}{4}f'\left(\frac{2\upsilon+\varphi}{3}\right)+\frac{3h^2}{4}f'\left(\frac{\upsilon+2\varphi}{3}\right)+\left[\frac{7h^5}{80}\right] f^4(\xi)$$

We finally have,

$$\mathrm{PS1(f{:}\,t)} = \frac{3h}{2}\big(f(x_0)+f(x_1)\big)-\frac{3}{4}h^2[f'(x_o)-f'(x_1)]+\frac{7h^5}{80}f^4(\xi)$$ ■

***Theorem* 2. 4** *Assuming that $f\ (t)$ and $g\ (t)$ are continuous on (υ, φ) and that $g(t)$ increases there. Let the interval (υ, φ) be divided into $2n$ subintervals (x_l, x_{l+1}) through equal-width $h = \frac{\varphi-\upsilon}{n+2}$ partition with quadrature nodes at $x_l = \upsilon + lh$, where $l = 0,1, \ldots, n$. Under these assumptions, the composite form of PS1, labelled as CPS1, to approximate the RSI takes the form:*

$$RS(f{:}\,g) \approx CPS1 = \sum_{l=1}^{n}\left[\begin{pmatrix} 5g(x_l)+4g(x_{l-1})-\frac{36}{\varphi-\upsilon}N_1(x_{l-1}\,,x_l) \\ +\frac{162}{(\varphi-\upsilon)^2}N_2(x_{l-1}\,,x_l)-\frac{324}{(\varphi-\upsilon)^3}N_3(x_{l-1}\,,x_l) \end{pmatrix} f\left(\frac{2x_{l-1}+x_l}{3}\right)\right]$$

$$+\sum_{l=1}^{n}\left[\begin{pmatrix} -4g(x_l)-5g(x_{l-1})+\dfrac{36}{\varphi-v}N_1(x_{l-1},x_l) \\ -\dfrac{162}{(\varphi-v)^2}N_2(x_{l-1},x_l)+\dfrac{324}{(\varphi-v)^3}N_3(x_{l-1},x_l)\end{pmatrix} f\left(\frac{x_{l-1}+2x_l}{3}\right)\right]$$

$$+\sum_{l=1}^{n}\left[\begin{pmatrix} \dfrac{2(\varphi-v)}{3}g(x_l)+\dfrac{4(\varphi-v)}{3}g(x_{l-1})-5N_1(x_{l-1},x_l) \\ +\dfrac{24}{\varphi-v}N_2(x_{l-1},x_l)-\dfrac{54}{(\varphi-v)^2}N_3(x_{l-1},x_l)\end{pmatrix} f'\left(\frac{2x_{l-1}+x_l}{3}\right)\right]$$

$$+\sum_{l=1}^{n}\left[\begin{pmatrix} \dfrac{4(\varphi-v)}{3}g(x_l)+\dfrac{2(\varphi-v)}{3}g(x_{l-1})-8N_1(x_{l-1},x_l) \\ +\dfrac{30}{\varphi-v}N_2(x_{l-1},x_l)-\dfrac{54}{(\varphi-v)^2}N_3(x_{l-1},x_l)\end{pmatrix} f'\left(\frac{x_{l-1}+2x_l}{3}\right)\right]$$

Proof of Theorem 2.4

Theorem 2.1 gives us:

$$PS1(f:g)=\begin{pmatrix} 5g(\varphi)+4g(\upsilon)-\dfrac{36}{\varphi-\upsilon}N_1(\upsilon,\varphi)+\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon,\varphi) \\ -\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon,\varphi)\end{pmatrix} f\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} -4g(\varphi)-5g(\upsilon) \\ +\dfrac{36}{\varphi-\upsilon}N_1(\upsilon,\varphi) \\ -\dfrac{162}{(\varphi-\upsilon)^2}N_2(\upsilon,\varphi) \\ +\dfrac{324}{(\varphi-\upsilon)^3}N_3(\upsilon,\varphi)\end{pmatrix} f\left(\frac{\upsilon+2\varphi}{3}\right) + \begin{pmatrix} \dfrac{2(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{4(\varphi-\upsilon)}{3}g(v) \\ -5N_1(\upsilon,\varphi)+\dfrac{24}{\varphi-\upsilon}N_2(\upsilon,\varphi) \\ -\dfrac{54}{(\varphi-\upsilon)^2}N_3(\upsilon,\varphi)\end{pmatrix} f'\left(\frac{2\upsilon+\varphi}{3}\right)$$

$$+\begin{pmatrix} \dfrac{4(\varphi-\upsilon)}{3}g(\varphi)+\dfrac{2(\varphi-\upsilon)}{3}g(\upsilon)-8N_1(\upsilon,\varphi) \\ +\dfrac{30}{\varphi-\upsilon}N_2(\upsilon,\varphi)\ -\dfrac{54}{(\varphi-\upsilon)^2}N_3(\upsilon,\varphi)\end{pmatrix} f'\left(\frac{\upsilon+2\varphi}{3}\right) \tag{2.29}$$

Applying rule on (2.29) over each sub-interval, we have:

$$CPS1(f:g)=\begin{pmatrix} 5g(x_1)+4g(\upsilon)-\dfrac{36}{\frac{\varphi-v}{n+2}}N_1(\upsilon,x_1) \\ +\dfrac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(\upsilon,x_1)-\dfrac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(\upsilon,x_1)\end{pmatrix} f\left(\frac{2\upsilon+x_1}{3}\right)$$

$$+\begin{pmatrix} -4g(x_1)-5g(\upsilon)+\dfrac{36}{\frac{\varphi-\chi}{n+2}}N_1(\upsilon,x_1) \\ -\dfrac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(\upsilon,x_1)+\dfrac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(\upsilon,x_1)\end{pmatrix} f\left(\frac{\upsilon+2x_1}{3}\right)$$

$$+\Bigg(\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_1)+\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(\upsilon)-5N_1(\upsilon\,,x_1)+\frac{24}{\frac{\varphi-\upsilon}{n+2}}N_2(\upsilon\,,x_1)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(\upsilon\,,x_1)\Bigg)f'\left(\frac{2\upsilon+x_1}{3}\right)+$$

$$\Bigg(\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_1)+\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(\upsilon)-8N_1(\upsilon\,,x_1)+\frac{30}{\frac{\varphi-\upsilon}{n+2}}N_2(\upsilon\,,x_1)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(\upsilon\,,x_1)\Bigg)f'\left(\frac{\upsilon+2x_1}{3}\right)+$$

$$\Bigg(5g(x_2)+4g(x_1)-\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_1\,,x_2)+\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_1\,,x_2)-\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_1\,,x_2)\Bigg)f\left(\frac{2x_1+x_2}{3}\right)+$$

$$\Bigg(-4g(x_2)-5g(x_1)+\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_1\,,x_2)-\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_1\,,x_2)+\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_1\,,x_2)\Bigg)f\left(\frac{x_1+2x_2}{3}\right)\ +$$

$$\Bigg(\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_2)+\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_1)-5N_1(x_1\,,x_2)+\frac{24}{\frac{\varphi-\upsilon}{n+2}}N_2(x_1\,,x_2)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_1\,,x_2)\Bigg)f'\left(\frac{2x_1+x_2}{3}\right)+$$

$$\Bigg(\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_2)+\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_1)-8N_1(x_1\,,x_2)+\frac{30}{\frac{\varphi-\upsilon}{n+2}}N_2(x_1\,,x_2)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_1\,,x_2)\Bigg)f'\left(\frac{x_1+2x_2}{3}\right)+$$

$$\cdots\Bigg(5g(x_l)+4g(x_{l-1})-\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_{l-1}\,,x_l)+\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_{l-1}\,,x_l)-\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_{l-1}\,,x_l)\Bigg)f\left(\frac{2x_{l-1}+x_l}{3}\right)+$$

$$\Bigg(-4g(x_l)-5g(x_{l-1})+\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_{l-1}\,,x_l)-\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_{l-1}\,,x_l)+\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_{l-1}\,,x_l)\Bigg)f\left(\frac{x_{l-1}+2x_l}{3}\right)\ +$$

$$\Bigg(\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_l)+\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_{l-1})-5N_1(x_{l-1}\,,x_l)+\frac{24}{\frac{\varphi-\upsilon}{n+2}}N_2(x_{l-1}\,,x_l)-$$

$$\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_{l-1}\,,x_l)\Bigg)f'\left(\frac{2x_{l-1}+x_l}{3}\right)+\Bigg(\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_l)+\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_{l-1})-8N_1(x_{l-1}\,,x_l)+$$

$$\frac{30}{\frac{\varphi-\upsilon}{n+2}}N_2(x_{l-1}\,,x_l)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_{l-1}\,,x_l)\Bigg)f'\left(\frac{x_{l-1}+2x_l}{3}\right)+\ldots\Bigg(5g(\varphi)+4g(x_{n-1})-\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_{n-1}\,,\varphi)+$$

$$\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_{n-1}\,,\varphi)-\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_{n-1}\,,\varphi)\Bigg)f\left(\frac{2x_{n-1}+\varphi}{3}\right)+\Bigg(-4g(\varphi)-5g(x_{n-1})+\frac{36}{\frac{\varphi-\upsilon}{n+2}}N_1(x_{n-1}\,,\varphi)-$$

$$\frac{162}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_2(x_{n-1}\,,\varphi)+\frac{324}{\left(\frac{\varphi-\upsilon}{n+2}\right)^3}N_3(x_{n-1}\,,\varphi)\Bigg)f\left(\frac{x_{n-1}+2\varphi}{3}\right)$$

$$+\begin{pmatrix} \frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(\varphi)+\frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_{n-1})-5N_1(x_{n-1},\varphi) \\ +\frac{24}{\frac{\varphi-\upsilon}{n+2}}N_2(x_{n-1},\varphi)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_{n-1},\varphi) \end{pmatrix} f'\left(\frac{2x_{n-1}+\varphi}{3}\right)$$

$$+\begin{pmatrix} \frac{4\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(\varphi)+\frac{2\left(\frac{\varphi-\upsilon}{n+2}\right)}{3}g(x_{n-1})-8N_1(x_{n-1},\varphi) \\ +\frac{30}{\frac{\varphi-\upsilon}{n+2}}N_2(x_{n-1},\varphi)-\frac{54}{\left(\frac{\varphi-\upsilon}{n+2}\right)^2}N_3(x_{n-1},\varphi) \end{pmatrix} f'\left(\frac{x_{n-1}+2\varphi}{3}\right)$$

Or, we can write in the compact form as:

$$CPS1(f:g)=\sum_{l=1}^{n}\left\{\begin{pmatrix} 5g(x_l)+4g(x_{l-1})-\frac{36(n+2)}{\varphi-\upsilon}N_1(x_{l-1},x_l) \\ +\frac{162(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1},x_l)-\frac{324(n+2)^3}{(\varphi-\upsilon)^3}N_3(x_{l-1},x_l) \end{pmatrix} f\left(\frac{2x_{l-1}+x_l}{3}\right)\right.$$

$$\left.+\begin{pmatrix} -4g(x_l)-5g(x_{l-1})+\frac{36(n+2)}{\varphi-\upsilon}N_1(x_{l-1},x_l) \\ -\frac{162(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1},x_l)+\frac{324}{(\varphi-\upsilon)^3}N_3(x_{l-1},x_l) \end{pmatrix} f\left(\frac{x_{l-1}+2x_l}{3}\right)\right\}$$

$$+\sum_{k=1}^{n}\left\{\begin{pmatrix} \frac{2(\varphi-\upsilon)}{3(n+2)}g(x_l)+\frac{4(\varphi-\upsilon)}{3(n+2)}g(x_{l-1})-5N_1(x_{l-1},x_l) \\ +\frac{24(n+2)}{\varphi-\upsilon}N_2(x_{l-1},x_l)-\frac{54(n+2)^2}{(\varphi-\upsilon)^2}N_3(x_{l-1},x_l) \end{pmatrix} f'\left(\frac{2x_{l-1}+x_l}{3}\right)\right.$$

$$\left.+\begin{pmatrix} \frac{4(\varphi-\upsilon)}{3(n+2)}g(x_k)+\frac{2(\varphi-\upsilon)}{3(n+2)}g(x_{k-1})-8N_1(x_{l-1},x_l) \\ +\frac{30(n+2)}{\varphi-\upsilon}N_2(x_{l-1},x_l)-\frac{54(n+2)^2}{(\varphi-\upsilon)^2}N_3(x_{l-1},x_l) \end{pmatrix} f'\left(\frac{x_{l-1}+2x_l}{3}\right)\right\}\blacksquare$$

***Theorem* 2.5:** *Assuming that $f(t)$ and $g(t)$ be continuous on $(\upsilon,\ \varphi)$ and $g(t)$ be increasing there. Let the interval $(\upsilon,\ \varphi)$ be divided into $2n$ subintervals (x_l, x_{l+1}) with width $h=\frac{\varphi-\upsilon}{n+2}$ partition with quadrature nodes at $x_l=\upsilon+lh$, where $l=0,1,\ldots,n$ under these assumptions, the global form error of the CPS1 takes the form:*

$R_{CPS1} = CPS1 + R_{CPS1}$

$$
=\sum_{l=1}^{n}\left\{\begin{array}{l}
\begin{pmatrix} 5g(x_l)+4g(x_{l-1})-\dfrac{36(n+2)}{\varphi-\upsilon}N_1(x_{l-1}\,,x_l) \\ +\dfrac{162(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1}\,,x_l)-\dfrac{324(n+2)^3}{(\varphi-\upsilon)^3}N_3(x_{l-1}\,,x_l)\end{pmatrix} f\left(\dfrac{2x_{l-1}+x_l}{3}\right) \\
+\begin{pmatrix} -4g(x_l)-5g(x_{l-1})+\dfrac{36(n+2)}{\varphi-\upsilon}N_1(x_{l-1}\,,x_l) \\ -\dfrac{162(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1}\,,x_l)+\dfrac{324}{(\varphi-\upsilon)^3}N_3(x_{l-1}\,,x_l)\end{pmatrix} f\left(\dfrac{x_{l-1}+2x_l}{3}\right) \\
+\begin{pmatrix} \dfrac{2(\varphi-\upsilon)}{3(n+2)}g(x_l)+\dfrac{4(\varphi-\upsilon)}{3(n+2)}g(x_{l-1})-5N_1(x_{l-1}\,,x_l) \\ +\dfrac{24(n+2)}{\varphi-\upsilon}N_2(x_{l-1}\,,x_l)-\dfrac{54(n+2)^2}{(\varphi-\upsilon)^2}N_3(x_{l-1}\,,x_l)\end{pmatrix} f'\left(\dfrac{2x_{l-1}+x_l}{3}\right) \\
+\begin{pmatrix} \dfrac{4(\varphi-\upsilon)}{3(n+2)}g(x_l)+\dfrac{2(\varphi-\upsilon)}{3(n+2)}g(x_{l-1})-8N_1(x_{l-1}\,,x_l) \\ +\dfrac{30(n+2)}{\varphi-\upsilon}N_2(x_{l-1}\,,x_l)-\dfrac{54(n+2)^2}{(\varphi-\upsilon)^2}N_3(x_{l-1}\,,x_l)\end{pmatrix} f'\left(\dfrac{x_{k-1}+2x_k}{3}\right)
\end{array}\right\}
$$

$$
+n\begin{pmatrix} \dfrac{(3\varphi^2-\upsilon^2-2\upsilon\varphi)}{2}g(\varphi)+\dfrac{(\varphi^2-3\upsilon^2+2\upsilon\varphi)}{2}g(\upsilon) \\ -\varphi N_1(\upsilon\,,\varphi)+N_2(\upsilon\,,\varphi)\end{pmatrix} f^{(4)}(\mu)g'(\eta)\ (2.30)
$$

***Proof of Theorem* 2.5**

Summing over the index m for n times the basic form error term in a particular subinterval $[x_{l-1},x_l]$, we can write:

$$
R_{CPS1}=\sum_{l=1}^{n}\begin{bmatrix} \dfrac{(x_l-x_{l-1})^4}{486}\ g(x_l)-\dfrac{(\varphi-\upsilon)^4}{486}g(x_{l-1})-\dfrac{(\varphi-\upsilon)^3}{54}N_1(x_{l-1}\,,x_l) \\ +\dfrac{13(\varphi-\upsilon)^2}{108}N_2(x_{l-1}\,,x_l)-\dfrac{(\varphi-\upsilon)}{2}N_3(x_{l-1}\,,x_l)+N_4(x_{l-1}\,,x_l)\end{bmatrix} f^{(4)}(\mu)g'(\eta)\ (2.31)
$$

Therefore, the global error is calculated by adding up n of these terms, and simplifying (2.31) gives:

$$
R_{CPS1}=n\begin{pmatrix} \frac{(\varphi-\upsilon)^4}{486}\ g(\varphi)-\frac{(\varphi-\upsilon)^4}{486}g(\upsilon)-\frac{(\varphi-\upsilon)^3}{54}N_1(\upsilon\,,\varphi) \\ +\frac{13(\varphi-\upsilon)^2}{108}N_2(\upsilon\,,\varphi)-\frac{(\varphi-\upsilon)}{2}N_3(\upsilon\,,\varphi)+N_4(\upsilon\,,\varphi)\end{pmatrix}\left(\tfrac{1}{n}\textstyle\sum_{l=1}^{n}f^{(4)}(\mu)g'(\eta)\right)
$$

Let $0=\dfrac{1}{n}\displaystyle\sum_{L=1}^{n}f^{(4)}(\mu)g'(\eta)$

Clearly $\min\limits_{x\in[\upsilon\,,\,\varphi]}\{f^{(4)}(x)g'(x)\}\le O\le\max\limits_{x\in[\upsilon\,,\,\varphi]}\{f^{(4)}(x)g'(x)\}$. Since $f^{(4)}(t)$ and $g'(t)$ are continuous in$[\upsilon\,,\ \varphi]$, then there exist two points $\mu\,,\eta$ such that $O=f^{(4)}(\mu)g'(\eta)$

This shows that the error term R_{CPS1} is

$$R_{CPS1} = n\begin{pmatrix} \dfrac{(\varphi-\upsilon)^4}{486}\; g(\varphi) - \dfrac{(\varphi-\upsilon)^4}{486} g(\upsilon) - \dfrac{(\varphi-\upsilon)^3}{54} N_1(\upsilon\,,\varphi) \\ + \dfrac{13(\varphi-\upsilon)^2}{108} N_2(\upsilon\,,\varphi) - \dfrac{(\varphi-\upsilon)}{2} N_3(\upsilon\,,\varphi) + N_4(\upsilon\,,\varphi) \end{pmatrix} f^{(4)}(\mu) g'(\eta)$$

Where μ and $\eta \in [\upsilon\,,\ \varphi]$ $and\ h = \frac{\varphi-\upsilon}{n+2}$. ■

In this manner, the basic and composite form RSI approximations using the third two-point proposed open-type schemes, respectively PS2 and CPS2, are stated as in (2.32) and (2.33). The corresponding error terms are of order 7 and 6 in (2.32) and (2.33), respectively.

$$\begin{aligned} RSI(f\colon g) \;=\; PS2 \;+\; R_{PS2}[f] \;=& \begin{pmatrix} -\dfrac{7}{2} g(\varphi) + \dfrac{9}{2} g(\upsilon) + \dfrac{39}{\varphi - v} N_1(\upsilon\,,\varphi) \\ -\dfrac{304}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) + \dfrac{1440}{(\varphi-\upsilon)^3} N_3(\upsilon\,,\varphi) \\ -\dfrac{3072}{(\varphi-\upsilon)^4} N_4(\upsilon\,,\varphi) \end{pmatrix} f\left(\frac{3\upsilon+\varphi}{4}\right) \\ &+ \begin{pmatrix} 9g(\varphi) - 9g(\upsilon) - \dfrac{96}{\varphi-\upsilon} N_1(\upsilon\,,\varphi) \\ +\dfrac{704}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \\ -\dfrac{3072}{(\varphi-\upsilon)^3} N_3(\upsilon\,,\varphi) \\ +\dfrac{6144}{(\varphi-\upsilon)^4} N_4(\upsilon\,,\varphi) \end{pmatrix} f\left(\frac{\upsilon+\varphi}{2}\right) + \begin{pmatrix} -\dfrac{9}{2} g(\varphi) + \dfrac{7}{2} g(\upsilon) \\ +\dfrac{57}{\varphi-\upsilon} N_1(\upsilon\,,\varphi) \\ -\dfrac{400}{(\varphi-\upsilon)^2} N_2(\upsilon\,,\varphi) \\ +\dfrac{1632}{(\varphi-\upsilon)^3} N_3(\upsilon\,,\varphi) \\ -\dfrac{3072}{(\varphi-\upsilon)^4} N_4(\upsilon\,,\varphi) \end{pmatrix} f\left(\frac{\upsilon+3\varphi}{4}\right) \\ &+ \begin{pmatrix} \dfrac{9(\varphi-\upsilon)}{8} g(\varphi) - \dfrac{3(\varphi-\upsilon)}{8} g(\upsilon) - \dfrac{39}{4} N_1(\upsilon\,,\varphi) \\ +\dfrac{58}{\varphi-\upsilon} N_2(\upsilon\,,\varphi) - \dfrac{216}{(\varphi-\upsilon)^2} N_3(\upsilon\,,\varphi) + \dfrac{384}{(\varphi-\upsilon)^3} N_4(\upsilon\,,\varphi) \end{pmatrix} f'\left(\frac{\upsilon+3\varphi}{4}\right) \\ &+ \begin{pmatrix} -\dfrac{3(\varphi-\upsilon)}{8} g(\varphi) + \dfrac{9(\varphi-\upsilon)}{8} g(\upsilon) + \dfrac{17}{4} N_1(\upsilon\,,\varphi) \\ -\dfrac{34}{\varphi-\upsilon} N_2(\upsilon\,,\varphi) + \dfrac{168}{(\varphi-\upsilon)^2} N_3(\upsilon\,,\varphi) - \dfrac{384}{(\varphi-\upsilon)^3} N_4(\upsilon\,,\varphi) \end{pmatrix} f'\left(\frac{3\upsilon+b}{4}\right) + \; R_{PS2}[f] \qquad (2.32) \end{aligned}$$

$$RSI(f\colon g) \;=\; CPS2 \;+\; R_{CPS2}[f]$$

$$\sum_{l=1}^{n}\left[\begin{array}{l}\left(\begin{array}{c}-\frac{7}{2}g(x_l)+\frac{9}{2}g(x_{l-1})+\frac{39(n+2)}{\varphi-\upsilon}N_1(x_{l-1},x_l)\\ -\frac{304(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1},x_l)+\frac{1440(n+2)^3}{(\varphi-\upsilon)^3}N_3(x_{l-1},x_l)\\ -\frac{3072(n+2)^4}{(\varphi-\upsilon)^4}N_4(x_{l-1},x_l)\end{array}\right)f\left(\frac{3x_{l-1}+x_l}{4}\right)\\ +\left(\begin{array}{c}9g(x_l)-9g(x_{l-1})-\frac{96(n+2)}{\varphi-\upsilon}N_1(x_{l-1},x_l)\\ +\frac{704(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1},x_l)-\frac{3072(n+2)^3}{(\varphi-\upsilon)^3}N_3(x_{l-1},x_l)\\ +\frac{6144(n+2)^4}{(\varphi-\upsilon)^4}N_4(x_{l-1},x_l)\end{array}\right)f\left(\frac{x_{l-1}+x_l}{2}\right)\\ +\left(\begin{array}{c}-\frac{9}{2}g(x_l)+\frac{7}{2}g(x_{l-1})+\frac{57(n+2)}{\varphi-\upsilon}N_1(x_{l-1},x_l)\\ -\frac{400(n+2)^2}{(\varphi-\upsilon)^2}N_2(x_{l-1},x_l)+\frac{1632(n+2)^3}{(\varphi-\upsilon)^3}N_3(x_{l-1},x_l)\\ -\frac{3072(n+2)^4}{(\varphi-\upsilon)^4}N_4(x_{l-1},x_l)\end{array}\right)f\left(\frac{x_{l-1}+3x_l}{4}\right)\end{array}\right]$$

$$+\sum_{l=1}^{n}\left[\left(\begin{array}{c}-\frac{3(\varphi-\upsilon)}{8(n+2)}g(x_l)+\frac{9(\varphi-\upsilon)}{8\,(n+2)}g(x_{l-1})+\frac{17}{4}N_1(x_{l-1},x_l)\\ -\frac{34(n+2)}{\varphi-\upsilon}N_2(x_{l-1},x_l)+\frac{168(n+2)^2}{(\varphi-\upsilon)^2}N_3(x_{l-1},x_l)\\ -\frac{384(n+2)^3}{(\varphi-\upsilon)^3}N_4(x_{l-1},x_l)\end{array}\right)f'\left(\frac{3x_{l-1}+x_l}{4}\right)\right]+R_{CPS2}[f](2.33)$$

3. Comparison of theoretical properties of proposed and some existing RSI approximations

In Table 1, the summary of theoretical parameters of the proposed PS1-PS2 and some existing Mercer, ZCT, MZCT, CS13, GMS13, HeMS13, CMS13, and HMS13 [16], [6], [7], [14], [20], [26], [24], and [15] existing schemes RSI approximations are mentioned. The precision degree, orders of accuracy in basic and global form approximations are mentioned. The total number of integral$(M_1, M_2, M_{3,}M_{4,})$, derivative $(f\,', f'')$, and function (f, g) type information used for all proposed and some existing RSI approximation in composite fashion with m −strip schemes are displayed in Table 2. Besides the particular amount of information used is also mentioned for the basic case m = 1 for the single execution of the formulas. In practice, these methods need varying number of repeated executions m to achieve a pre-specified upper limit on the absolute error; hence the performance in terms of total computational expense varies particularly from problem to problem.

Table 1: Summary of theoretical parameters of the proposed RSI approximations

RSI approximations	Local order of accuracy	Global order of accuracy	Precision
PS1	5	4	3
PS2	7	6	5

Table 2: Computational expense breakdown for m-strip execution in RSI approximations

Information	f	g	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$	$f^{(4)}$	M_1	M_2	M_3	M_4	Total m=1
CT	$m+1$	2	0	0	0	0	m	0	0	0	$2m+3$
ZCT	$m+1$	2	0	m	0	0	m	m	m	0	$5m+3$
MZCT	$m+1$	2	0	m	0	0	m	m	m	0	$5m+3$
PS1	$2m$	$m+1$	2m	0	0	0	m	m	m	0	$8m+1$
PS2	$3m$	$m+1$	2m	0	0	0	m	m	m	m	$10m+1$
CS13	$2m+1$	2	0	0	0	0	m	m	0	0	$4m+3$
GMS13	$2m+1$	2	0	0	0	m	m	m	m	m	$7m+3$
HeMS13	$2m+1$	2	0	0	0	m	m	m	m	m	$7m+3$
CMS13	$2m+1$	2	0	0	0	m	m	m	m	m	$7m+3$
HMS13	$2m+1$	2	0	0	0	m	m	m	m	m	$7m+3$

3. Numerical experiments, results and discussion

For each scheme selected from [26, 27], the present study resolved six numerical problems in order to evaluate the effectiveness of both current and suggested RSI approximations. We used a laptop with an Intel(R) Core(TM) processor operating at 1-1.61 GHz with 8GB of RAM to program and run the techniques for all the results we obtained. MATLAB R2019a was used to compute all of the results.

The problems 1-6 are listed here with their exact results from [26, 27].

Example 1: $\int_5^6 e^x \, dsinx = 1.8743E+02$ [26]

Example 2: $\int_{-1}^1 x^7 de^x = 2.5383E-01$ [26]

Example 3: $\int_0^{(\frac{\pi}{2})} x^7 dsinx = 7.9077E-01$ [27]

Example 4: $\int_5^6 sin5x dx^3 = 1.0232E+01$[27]

Example 5: $\int_{3.5}^{4.5} sin5x \, dcosx = 2.2768E-01$[27]

Example 6: $\int_{0.1}^{0.5} \frac{1}{e^{\frac{1}{x}}+1} d(\sin(x)) = 0.0159062$ [27]

These problems include a range of integrators, integrands, and limits of integrals which satisfy the standards of the RSI, allowing for a thorough comparison of the effectiveness of suggested and current RSI techniques. Comparing the performance of the proposed RSI schemes to that of the CT, ZCT, MZCT, CS13, GMS13, HeMS13, CMS13, and HMS13 [16], [6], [7], [14], [20], [26], [24], and [15] existing schemes. In [4], the absolute error is written as

$$Absolute\ Error\ = |\,Exact\ value\ -\ Approximate\ value\,| \qquad (3.1)$$

According to [28], the definition of the computational order of accuracy is

$$\text{Computational order of accuracy} = \rho = \frac{\ln\left(\frac{|N(2h)-N(0)|}{|N(h)-N(0)|}\right)}{\ln 2} \qquad (3.2)$$

Where N (h) represents the approximate value at step size h and N (0) represents the exact result. The line plots in Figs. 1–6 showed that, for all test problems, the proposed schemes' errors decreased more quickly than those of the existing schemes. The computational orders of accuracy for all schemes for all test problems are similar to those of the theoretically proved previous section, as shown in Tables 3, 4, 5, 6, 7, and 8. The proposed PS1 method has an order of accuracy of 4, the same as the MZCT, CS13, GMCS13, HeMCS13, CMCS13, and HMCS13 schemes; however, the former's error reduction process is quicker. The accuracy of the proposed PS2 schemes is 6. Due to the problems and errors mentioned in [7], the ZCT scheme has an order that varies and doesn't converge to 4. Table 2 shows the total number of function, derivative, and integral estimations for all m-strip schemes. Figs. 7 and 8 show computational performance in terms of total computational cost and the average CPU usage in seconds for the three integrals mentioned in Examples 1–6 using the CT, ZCT, MZCT, PS1, PS2, CS13, GMCS13, HeMCS13, CMCS13, and HMCS13 schemes. The numerical results show that, for all test problems, the proposed scheme PS2 required less cost to achieve the error compared to the existing schemes, as shown in Figs 7 and 8, which show that, in comparison to the existing schemes for Examples 1–6, the proposed scheme PS1 required less average CPU time to achieve the errors.

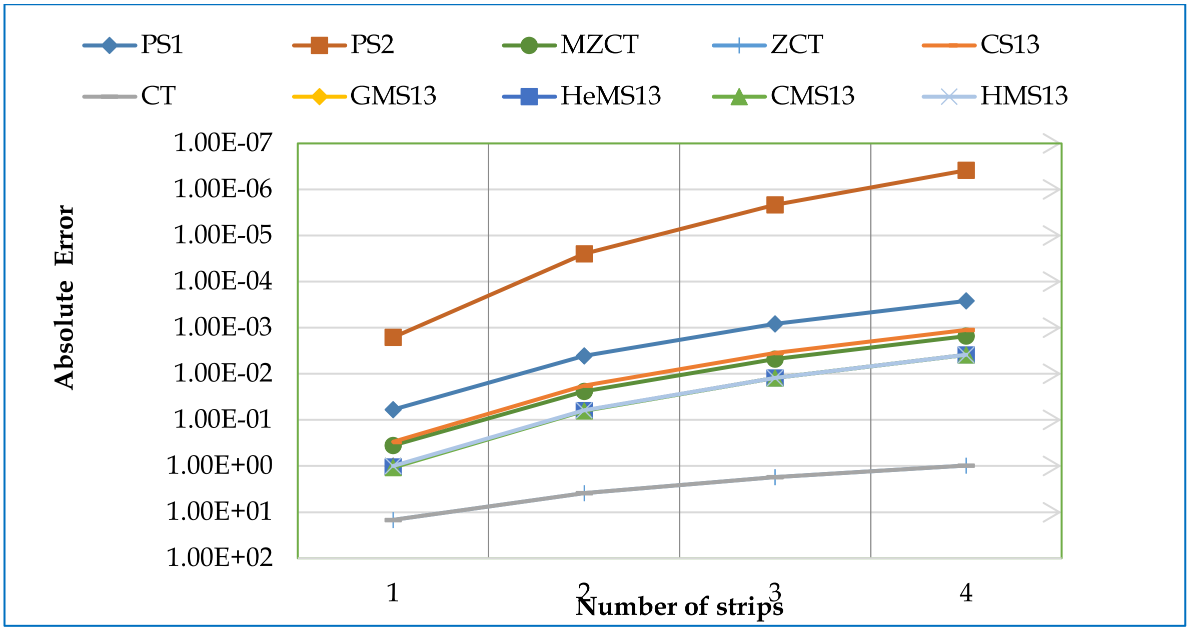

Fig 1. Plots of errors to estimate the integral in an example 1

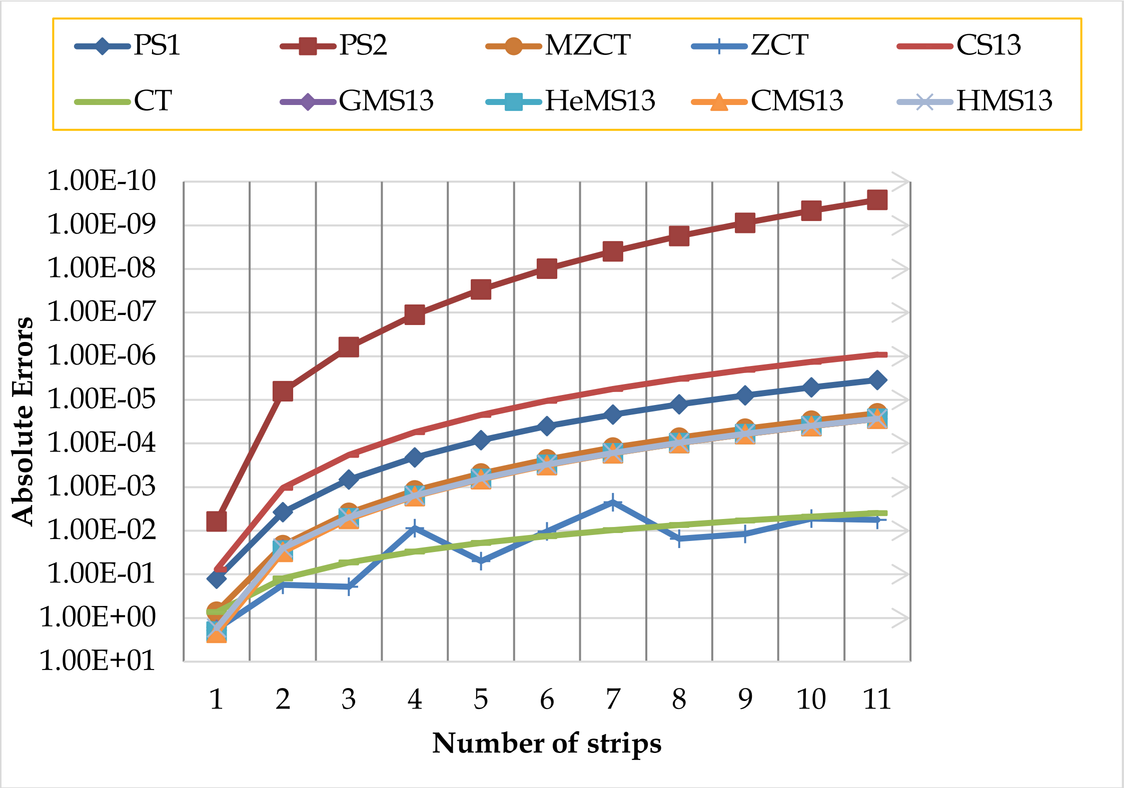

Fig 2. Plots of errors to estimate the integral in an example 2

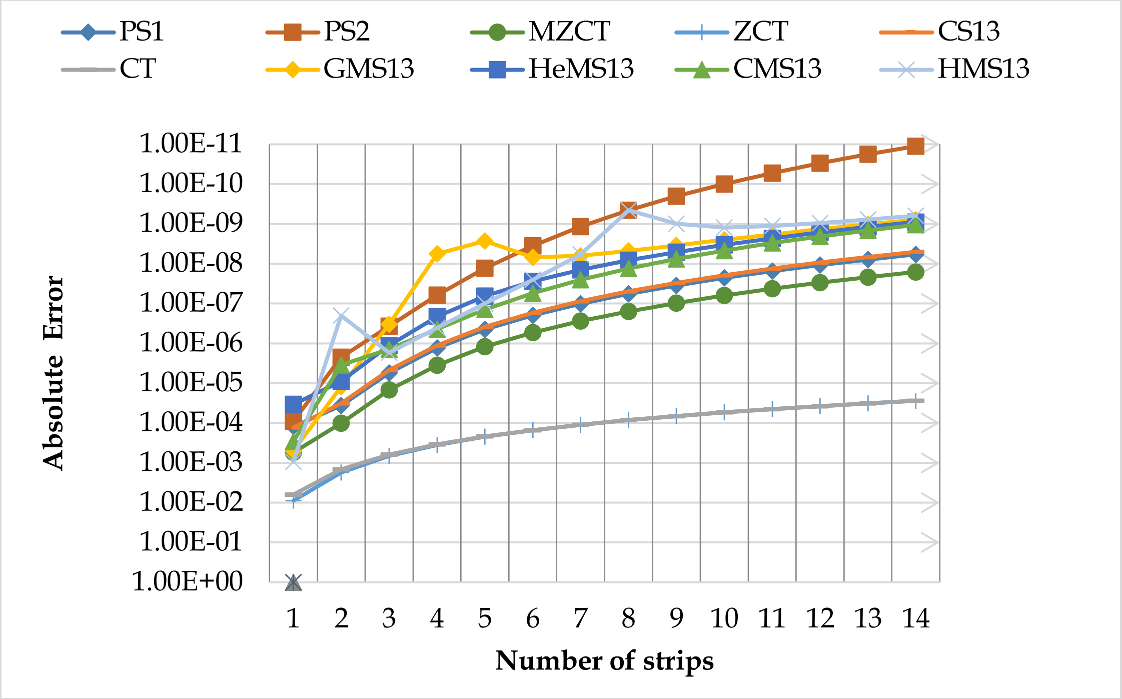

Fig 3. Plots of errors to estimate the integral in an example 3

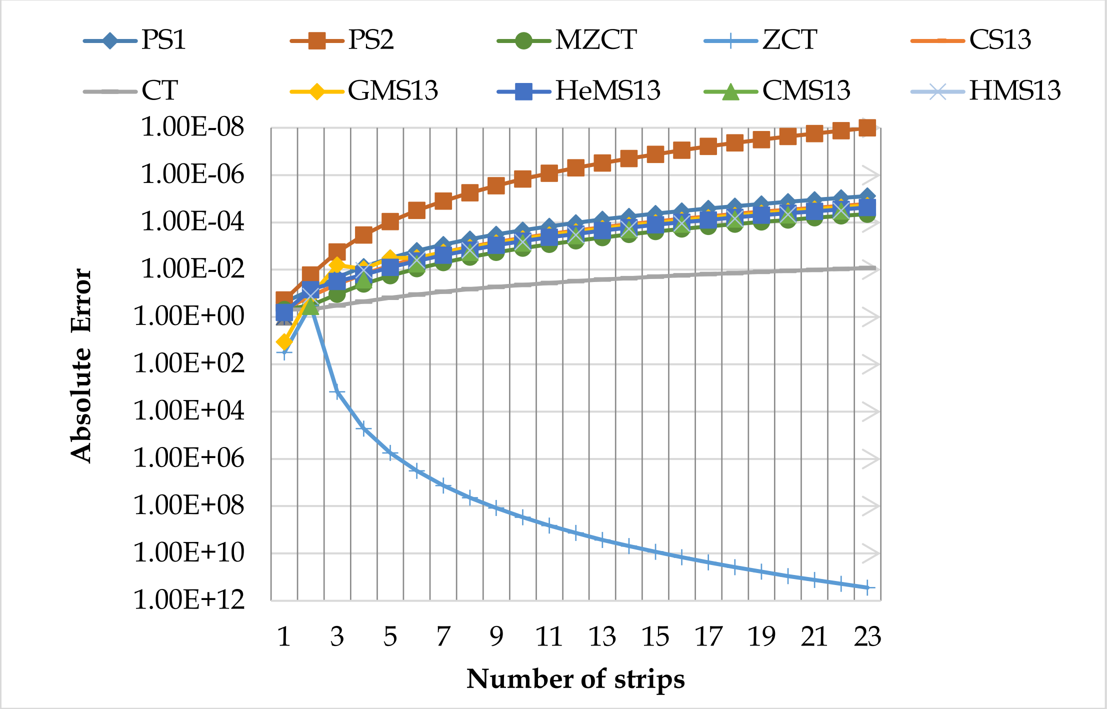

Fig 4 Plots of errors to estimate the integral in an example 4

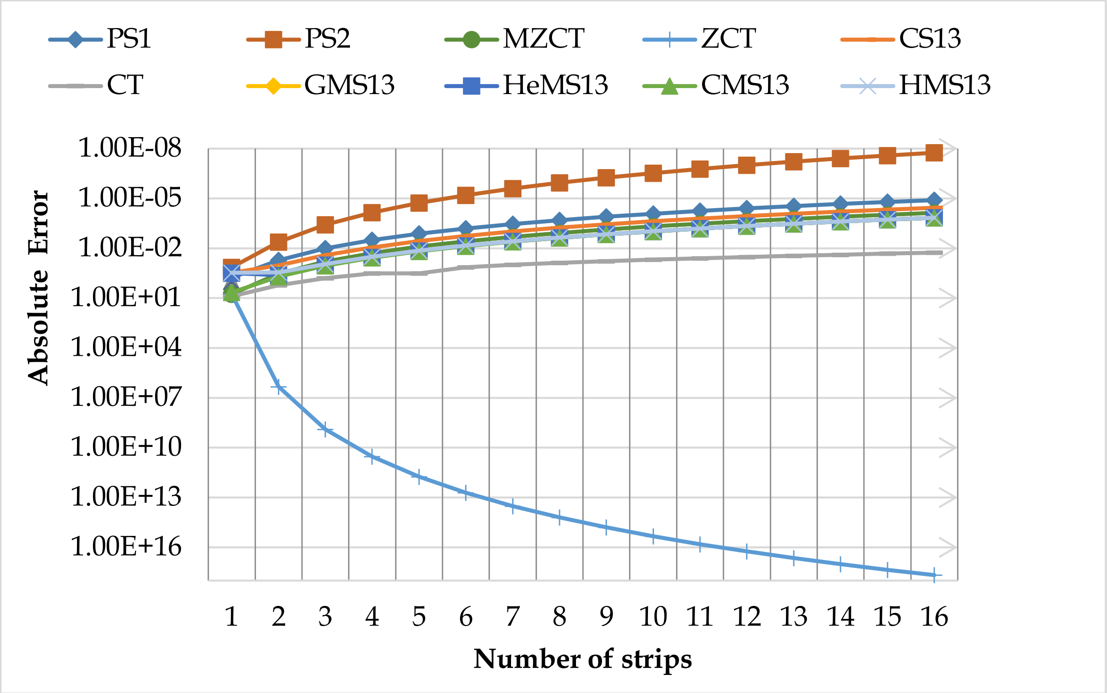

Fig 5 Plots of errors to estimate the integral in an example 5

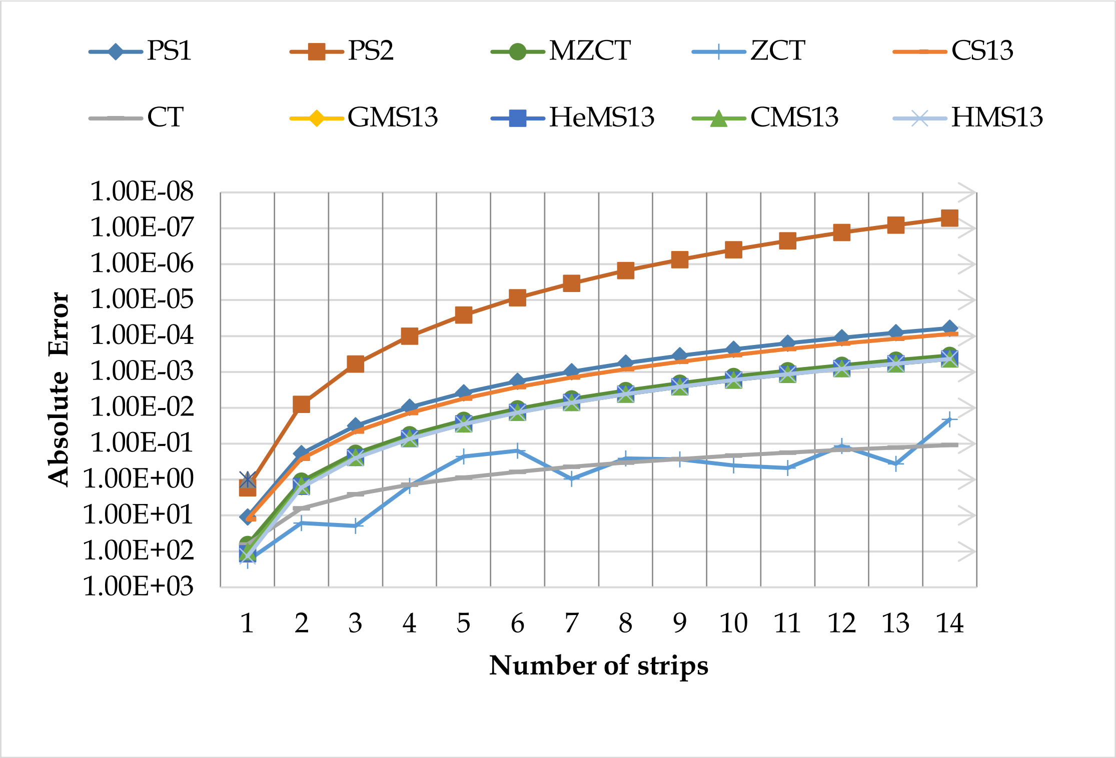

Fig 6 Plots of errors to estimate the integral in an example 6

Table 3: Computational order of accuracy comparison in Example 1

Repetitions m	**1**	**2**	**4**	**8**	**16**	**32**	**64**
PS1	NA	3.8618	3.9662	3.9916	3.9979	3.9995	4.0001
PS2	NA	6.0277	6.0085	6.0022	6.0011	Exact	Exact
CT	NA	1.9319	1.9844	1.9962	1.9990	1.9998	Exact
ZCT	NA	1.9319	1.9844	1.9962	1.9990	1.9998	Exact
MZT	NA	3.8984	3.9761	3.9941	3.9985	3.9996	Exact
CS13	NA	4.0475	4.0133	4.0034	4.0009	4.0002	Exact
GMS13	NA	4.0395	4.0134	4.0036	4.0009	4.0003	Exact
HeMS13	NA	4.0606	4.0187	4.0049	4.0012	4.0003	Exact
CMS13	NA	4.0921	4.0266	4.0069	4.0018	4.0004	Exact
HMS13	NA	4.0077	4.0054	4.0016	4.0004	4.0001	Exact

Table 4: Computational order of accuracy comparison in Example 2

Repetitions m	1	2	4	8	16	32	64
PS1	NA	1.7316	3.2889	3.8446	3.9624	3.9906	3.9977
PS2	NA	3.5046	5.6409	5.9167	5.9795	5.9949	5.9987
CT	NA	0	1.1309	1.7270	1.9298	1.9823	1.9956
ZCT	NA	6.6075	−10.725	−9.6717	−8.3922	- -	- -
MZT	NA	0.58236	2.9940	3.7856	3.9482	3.9871	Exact
CS13	NA	1.8756	3.1237	3.7911	3.9484	3.9871	Exact
GMS13	NA	6.4445	3.8106	3.0418	3.8139	3.9557	Exact
HeMS13	NA	3.1987	2.1137	3.4649	3.8830	3.9716	Exact
CMS13	NA	0	3.7702	3.8589	3.9609	3.9900	Exact
HMS13	NA	0	3.6438	2.7775	3.7711	3.9457	Exact

Table 5: Computational order of accuracy comparison in Example 3

Repetitions m	1	2	4	8	16	32	64
PS1	NA	3.4952	3.9944	4.0052	4.0017	4.0005	4.0001
PS2	NA	5.0904	5.8638	5.9691	5.9924	6.0086	Exact
CT	NA	2.2305	2.3322	2.1227	2.0341	2.0087	2.0022
ZCT	NA	−18.5393	−13.9355	−12.1748	−11.5229	−	-
MZT	NA	4.0678	4.1378	4.0385	4.0089	4.0019	Exact
CS13	NA	1.4456	3.6433	3.9215	3.9810	3.9953	3.9988
GMS13	NA	−0.2803	3.4255	3.8787	3.9709	3.9927	Exact
HeMS13	NA	−0.4519	3.5996	3.9121	3.9788	3.9947	Exact
CMS13	NA	3.1452	3.8243	3.9603	3.9903	3.9977	Exact
HMS13	NA	0.0614	3.1501	3.8281	3.9590	3.9898	Exact

Table 6: Computational order of accuracy comparison in Example 4

Repetitions m	1	2	4	8	16	32	64
PS1	NA	5.8525	4.2975	4.0688	4.0169	4.0042	4.0010
PS2	NA	7.7302	6.2946	6.0689	6.0170	6.0043	6.0010
CT	NA	3.3263	2.1920	2.0437	2.0107	2.0027	2.0007
ZCT	NA	3.5022	3.4089	2.5613	3.4993	0.2484	3.9568
MZT	NA	5.8741	4.3099	4.0722	4.0178	4.0044	Exact
CS13	NA	5.5955	4.2546	4.0588	4.0144	4.0036	4.0009
GMS13	NA	6.3037	4.4561	4.1112	4.0277	4.0069	Exact
HeMS13	NA	6.2463	4.4209	4.1009	4.0250	4.0063	Exact
CMS13	NA	6.1295	4.3636	4.0852	4.0210	4.0052	Exact
HMS13	NA	6.3647	4.5047	4.1263	4.0316	4.0079	Exact

Table 7: Computational order of accuracy comparison in Example 5

Repetitions m	1	2	4	8	16	32	64
PS1	NA	5.0647	4.1667	4.0397	4.0098	4.0024	4.0006
PS2	NA	9.8989	5.8351	5.9811	5.9962	6.0005	*Exact*
CT	NA	2.5728	2.0420	2.0091	2.0022	2.0006	2.0001
ZCT	NA	3.3788	4.3106	−0.8154	2.1713	3.2118	1.1185
MZT	NA	5.0724	4.1474	4.0348	4.0086	4.0021	4.0006
CS13	NA	6.1916	4.2504	4.0593	4.0147	4.0037	Exact
GMS13	NA	6.1615	4.0903	4.0206	4.0051	4.0013	Exact
HeMS13	NA	6.1797	4.1456	4.0337	4.0083	4.0021	Exact
CMS13	NA	6.1764	4.2210	4.0528	4.0131	4.0032	Exact
HMS13	NA	6.0987	3.9992	4.0006	4.0002	4.0000	Exact

Table 8: Computational order of accuracy comparison in Example 6

Repetitions m	1	2	4	8	16	32	64
PS1	NA	1.7385	4.8259	4.5037	4.0749	4.0133	4.0030
PS2	NA	5.3406	5.1929	7.0612	6.5690	6.1536	6.0346
CT	NA	2.0958	2.0984	2.0143	2.0026	2.0006	2.0002
ZCT	NA	2.3455	2.3030	2.0640	2.0054	2.0006	2.0002
MZT	NA	1.7516	4.8124	4.4937	4.0732	4.0129	4.0030
CS13	NA	2.0247	4.8403	4.4977	4.0749	4.0135	Exact
GMS13	NA	5.3515	11.080	0.23805	3.3161	–	–
HeMS13	NA	1.9248	5.3907	4.7047	3.9658	3.9800	Exact
CMS13	NA	6.3791	2.9931	5.0447	4.4622	4.1180	Exact
HMS13	NA	12.184	−1.0342	9.8126	0.15921	–	–

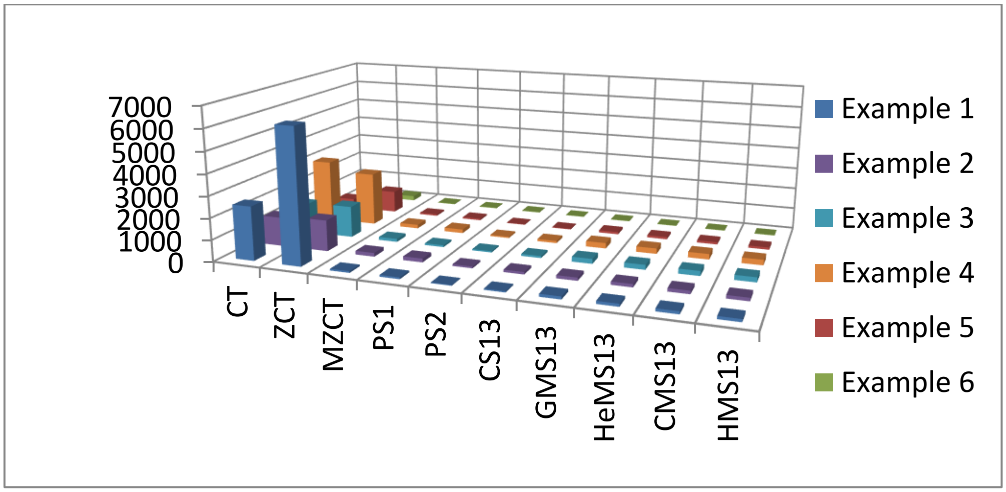

Figure 7: Computational cost comparison to achieve at most 1E-05 absolute error in quadrature variants for Examples 1-6

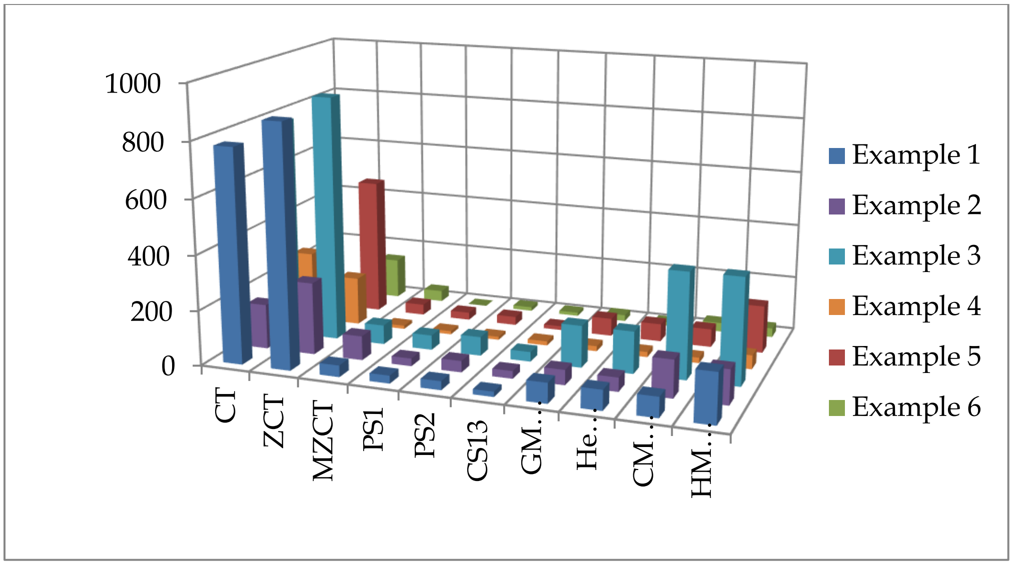

Figure 8: CPU time comparison to achieve at most 1E-05 absolute error in quadrature variants for Examples 1-6

4. Conclusion

Using the derivative information in the approximation process, a novel set of open-type RSI approximations was proposed in this study. In both the single and multiple execution cases, every result pertaining to the derivation and error analysis of the suggested RSI approximation has been demonstrated. Based on the characteristics of the suggested and some current RSI approximations from the literature, a theoretical comparison is also conducted. Six test RSIs from the literature were used to verify the theoretical parameters of the suggested RSI approximations. Additionally, some existing approximations based on standard indicators, such as drops of the absolute errors, observed/computational accuracy order, computational expense/ information used, and execution times, were compared to proposed results of RSI approximations.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S.D. Conte, C. de Boor, Elementary Numerical Analysis, SIAM, Philadelphia, 2017. https://doi.org/10.1137/1.9781611975208.

[2] R. Gordon, The Integrals of Lebesgue, Denjoy, Perron, and Henstock, American Mathematical Society, Providence, Rhode Island, 1994. https://doi.org/10.1090/gsm/004.

[3] P. Billingsley, Probability and Measure, John Wiley & Sons, (2017).

[4] F. Zafar, S. Saleem, C.O.E. Burg, New Derivative Based Open Newton-Cotes Quadrature Rules, Abstr. Appl. Anal. 2014 (2014), 109138. https://doi.org/10.1155/2014/109138.

[5] F.B. Hildebrand, Introduction to Numerical Analysis, Courier Corporation, (1987).

[6] W. Zhao, H. Li, Midpoint Derivative-Based Closed Newton-Cotes Quadrature, Abstr. Appl. Anal. 2013 (2013), 492507. https://doi.org/10.1155/2013/492507.

[7] M.M. Shaikh, M.S. Chandio, A.S. Soomro, A Modified Four-Point Closed Mid-Point Derivative Based Quadrature Rule for Numerical Integration, Sindh Univ. Res. J. (Sci. Ser.), 48 (2016), 389-392.

[8] W.H. Press, Numerical Recipes 3rd Edition, Cambridge University Press, (2007).

[9] W. Zhao, Z. Zhang, Z. Ye, Composite Trapezoid Rule for the Riemann-Stieltjes Integral and Its Richardson Extrapolation Formula, Italian J. Pure Appl. Math. 35 (2015), 311-318.

[10] W. Zhao, Z. Zhang, Derivative-Based Trapezoid Rule for the Riemann-Stieltjes Integral, Math. Probl. Eng. 2014 (2014), 874651. https://doi.org/10.1155/2014/874651.

[11] W. Zhao, Z. Zhang, Simpson's Rule for the Riemann-Stieltjes Integral, J. Interdiscip. Math. 24 (2021), 1305-1314. https://doi.org/10.1080/09720502.2020.1848317.

[12] W. Zhao, Z. Zhang, Derivative-Based Trapezoid Rule for a Special Kind of Riemann-Stieltjes Integral, Italian J. Pure Appl. Math. 50 (2023), 672–681.

[13] W. Zhao, Z. Zhang, Z. Ye, Midpoint Derivative-Based Trapezoid Rule for the Riemann-Stieltjes Integral, Italian J. Pure Appl. Math. 33 (2014), 369-376.

[14] K. Memon, A New and Efficient Simpson's 1/3-Type Quadrature Rule for Riemann-Stieltjes Integral, J. Mech. Contin. Math. Sci. 15 (2020), 132-148. https://doi.org/10.26782/jmcms.2020.11.00012.

[15] K. Memon, A New Harmonic Mean Derivative-Based Simpson's 1/3-Type Scheme for Riemann-Stieltjes Integral, J. Mech. Contin. Math. Sci. 16 (2021), 22-46. https://doi.org/10.26782/jmcms.2021.04.00003.

[16] M. Dehghan, M. Masjed-Jamei, M. Eslahchi, On Numerical Improvement of Open Newton–Cotes Quadrature Rules, Appl. Math. Comput. 175 (2006), 618-627. https://doi.org/10.1016/j.amc.2005.07.030.

[17] M. Dehghan, M. Masjed-Jamei, M. Eslahchi, The Semi-Open Newton–Cotes Quadrature Rule and Its Numerical Improvement, Appl. Math. Comput. 171 (2005), 1129-1140. https://doi.org/10.1016/j.amc.2005.01.137.

[18] M. Dehghan, M. Masjed-Jamei, M. Eslahchi, On Numerical Improvement of Closed Newton–Cotes Quadrature Rules, Appl. Math. Comput. 165 (2005), 251-260. https://doi.org/10.1016/j.amc.2004.07.009.

[19] E. Babolian, M. MasjedJamei, M. Eslahchi, On Numerical Improvement of Gauss–Legendre Quadrature Rules, Appl. Math. Comput. 160 (2005), 779-789. https://doi.org/10.1016/j.amc.2003.11.031.

[20] M.M. Shaikh, K. Memon, S. Mahesar, O.A. Rajput, A Theoretical Analysis of Simpson's 1/3-Type Scheme for the Riemann-Stieltjes Integral Based on Geometric Mean Derivative, Sindh Univ. Res. J. Sci. Ser. 57 (2025), 1-13. https://doi.org/10.26692/surjss.v57i1.7711.

[21] K. Memon, M.M. Shaikh, S. Mahesar, Numerical Investigation of a Four-Point Simpson's Quadrature for Computing Riemann-Stieltjes Integrals, NED Univ. J. Res. XXII (2025), 27-44. https://doi.org/10.35453/nedjr-ascn-2024-0017.r3.

[22] R. Marjulisa, M. Imran, Aturan Titik Tengah Double Untuk Mengaproksimasi Integral Riemann-Stieltjes, Jurnal Matematika Integratif, 19 (2023), 29–39.

[23] S. Mahesar, M.M. Shaikh, M.S. Chandio, A.W. Shaikh, Some New Time and Cost Efficient Quadrature Formulas to Compute Integrals Using Derivatives with Error Analysis, Symmetry 14 (2022), 2611. https://doi.org/10.3390/sym14122611.

[24] S. Mahesar, M.M. Shaikh, M.S. Chandio, A.W. Shaikh, Centroidal Mean Derivative-Based Open Newton-Cotes Quadrature Rules, VFAST Trans. Math. 11 (2023), 138-154. https://doi.org/10.21015/vtm.v11i2.1601.

[25] P.R. Mercer, Hadamard's Inequality and Trapezoid Rules for the Riemann–Stieltjes Integral, J. Math. Anal. Appl. 344 (2008), 921-926. https://doi.org/10.1016/j.jmaa.2008.03.026.

[26] K. Memon, Heronian Mean Derivative-Based Simpson's-Type Scheme for Riemann-Stieltjes Integral, J. Mech. Contin. Math. Sci. 16 (2021), 53-68. https://doi.org/10.26782/jmcms.2021.03.00005.

[27] R.L. Burden, J.D. Faires, Numerical Analysis, Brooks/Cole, Boston, (2011).

[28] M.M. Shaikh, Analysis of Polynomial Collocation and Uniformly Spaced Quadrature Methods for Second Kind Linear Fredholm Integral Equations–A Comparison, Turk. J. Anal. Number Theory 7 (2019), 91-97.