Journal of Informatics and Mathematical Sciences
Vol. 10, No. 3, pp. 429–442, 2018
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications

http://www.rgnpublications.com

DOI: 10.26713/jims.v10i3.773

Research Article

Advanced Family of Newton-Cotes Formulas

Mahesh Chalpuri[1,*], J. Sucharitha[1] and M. Madhu[2]

[1] *Department of Mathematics, Osmania University, Hyderabad 500007, India*
[2] *Department of Mathematics, University College of Palamuru University, Telangana 509001, India*
*__Corresponding author:__ mchalpuri@gmail.com

Abstract. In this paper, we introduced an advanced family of numerical composite integration formulas of closed Newton–Cotes-type that uses the function values on uniformly spaced intervals only without any derivative values. To increase the accuracy, we divide the given interval into a number of equal subintervals and integrating on each interval by using integration rules with abscissas outside integration interval. Since there are more unknowns when using including function values outside integration interval in addition to function values of on interval, the order of accuracy of these numerical integration formulas is higher than the standard closed Newton-Cotes formulae. These new formulae are obtained using the method of undetermined coefficients which are based on the concept of the precision of the quadrature formula. The error terms are found using the concept of precision. Also, we compared the errors in an advanced family of numerical composite integration formulas with the errors in composite closed Newton–Cotes-type. Finally, we have presented some examples and then mentioned the related MATLAB codes.

Keywords. Numerical Integration; Closed Newton-Cotes integration; Gauss Quadrature; Polynomial interpolation

MSC. 41A55; 65D30; 65D32

Received: January 29, 2018 **Accepted:** September 6, 2018

1. Introduction

If $f(x)$ is a real-valued function defined on the interval $[a,b]$, the Lagrange interpolation polynomial of degree n (or less) which agrees with $f(x)$ at the equidistant nodes $x_k = a + hk$, $k = 0,1,2,\dots,n$ with space $h = (b-a)/n$ will be denoted by $L_n(f,a,b;x)$. It is well-known [1,5] that

$$I[f]=\int_a^b f(x)dx=\int_a^b L_n(f,a,b;x)dx+E_{n+1}[f,a,b]$$
$$=\sum_{j=0}^{n} w_j f_j + E_{n+1}[f,a,b]$$

or

$$I[f]=I_{n+1}[f,a,b]+E_{n+1}[f,a,b], \tag{1}$$

where w_j's are weights, $f_j=f(x_j)$ for all j and $E_{n+1}[f,a,b]$ is error of numerical integration. this formula is often called the *Newton-Cotes integration* formula.

Using Taylor's series expansion of $f(x)$ about the midpoint of the interval, the relationship between the integral and the quadrature formula may be established for various quadrature formula. The actual order of accuracy of the resulting closed Newton-Cotes quadrature formula may be higher than expected, due to favorable cancellations. Several examples are shown below.

Trapezoid Rule ($n=1$)

$$I_2[f]=\frac{h}{2}(f_0+f_1)-\frac{h^3}{12}f''(\xi), \quad \xi\in(a,b). \tag{2}$$

Simpson's Rule ($n=2$)

$$I_3[f]=\frac{h}{3}(f_0+4f_1+f_2)-\frac{h^5}{90}f^{(4)}(\xi), \quad \xi\in(a,b). \tag{3}$$

Simpson's Three-Eighths Rule ($n=3$)

$$I_4[f]=\frac{3h}{8}(f_0+3f_1+3f_2+f_3)-\frac{3h^5}{80}f^{(4)}(\xi), \quad \xi\in(a,b). \tag{4}$$

Boole's Rule ($n=4$)

$$I_5[f]=\frac{2h}{45}(7f_0+32f_1+12f+32f_3+f_4)-\frac{8h^7}{945}f^{(6)}(\xi), \quad \xi\in(a,b). \tag{5}$$

These stencils can be generated quickly via mathematical software programs. From these results, one can observe that the order of accuracy for even n is $n+3$, whereas, the order of accuracy for odd n is only $n+2$. For composite quadrature formula for integrals over general intervals, using a multiple of n equally spaced subdivisions, the order of accuracy is reduced by one. The precision of a numerical integration scheme is directly related to the number of linearly independent equations that must be solved for the parameters within the scheme. For the Newton–Cotes methods, the function evaluations are uniformly spaced so that the weights are the only parameters to be determined. For Gauss–Legendre integration, both the locations and the weights need to be specified. Using the same number of function evaluations, the precision for the Gauss–Legendre integration is roughly twice that of the Newton-Cotes methods.

To avoid the use of higher order methods and still obtain accurate results, we use the composite integration methods. We divide the interval $[a,b]$ into a number of subintervals and evaluate the integral in each subinterval by a particular method. If we divide the interval $[a,b]$

into m equal subinterval namely A_j, $j = 1,\dots,m$. Then

$$I[f] = \sum_{j=1}^{m} \int_{A_j} f(x)dx. \tag{6}$$

Each interval can be divide into n equal subintervals and apply $n+1$ points *Newton-Cotes formula* in each integral, we get

$$I[f] = \sum_{i=1}^{m} (I_{n+1}[f, x_{i-1}, x_i] + E_{n+1}[f, x_{i-1}, x_i])$$

or simply

$$I[f] = I^c_{nm+1}[f] + E^c_{nm+1}[f] \tag{7}$$

this is called family of *composite closed Newton-Cotes* methods.

By taking of $n = 1$ in (7), we get

$$I^c_{m+1}[f] = \frac{(b-a)}{m}\left(\frac{1}{2}f(a) + \sum_{i=2}^{m-1} f(x_i) + \frac{1}{2}f(b)\right). \tag{8}$$

The sequence $I^c_2[f]$, $I^c_3[f]$,$I^c_4[f]\dots$ is called the *trapezoidal method* and the error term is

$$E^c_{m+1}[f] = -\frac{(b-a)^3}{12m^2} f''(\xi)$$

if f and $f'$are absolutely continuous and ξ in (a,b).

By taking of $n = 2$ in (7), we get

$$I^c_{2m+1}[f] = \frac{h}{3}(f(x_0) + 4f(x_1) + +2f(x_2) + \dots + 4f(x_{2m-1}) + f(x_{2m})). \tag{9}$$

The sequence $I^c_3[f]$,$I^c_5[f]$,$I^c_7[f]\dots$ is called the *Simpson method*. The error term of composite Simpson method is

$$E^c_{2m+1}[f] = -\frac{(b-a)^5}{180m^4} f^{(4)}(\xi)$$

if f and $f'''$are absolutely continuous and ξ in (a,b).

For a long time, Simpson's method was the most popular quadrature method, because it combines reasonable exactness with low computational cost (as it involves almost no multiplications). The importance of the latter advantage has diminished with the wide availability of fast computers, but Simpson's method retains its position as one of the classic and most frequently applied methods for numerical integration.

The trapezoidal method results from this approach and, while converging slowly itself. The Euler–Maclaurin formula gives rather precise information about $E_{n+1}[f,a,b]$. Gregory methods use this information to improve the trapezoidal rule forms the basis of the more powerful method of Gregory [1] is

$$I^{Gr,r}_{n+1}[f] = \sum_{v=0}^{n-r-1}\left(\int_{x_v}^{x_{v+1}} L_r(f, x_v, x_{v+r}; x)dx\right) + \int_{x_{n-r}}^{x_n} L_{r-1}(f, x_{n-r}, x_n; x)dx. \tag{10}$$

If $r = 1$ in (10) then the method is called *Durand method* [8] and it is

$$I^{Du}_{n+1}[f] = \frac{(b-a)}{n}\left(\frac{5}{12}f(x_0) + \frac{13}{12}f(x_1) + \sum_{i=2}^{n-1} f(x_i) + \frac{13}{12}f(x_{n-1}) + \frac{5}{12}f(x_n)\right) \tag{11}$$

which yields better results than the trapezoidal method, at least asymptotically when $f'(a) = f'(b)$ and order of (11) is 3.

One of the integral method with equidistant method is *Lacroix method* ($r = 2$ in (10)) and it is follows

$$I_{n+1}^{La} = \int_{x_0}^{x_1+h/2} L_2(f, x_0, x_2; x)dx + \sum_{i=1}^{n} \left(\int_{x_i-h/2}^{x_i+h/2} L_2(f, x_i, x_{i+2}; x)dx \right) + \int_{x_{n-1}+h/2}^{x_n} L_2(f, x_{n-2}, x_n; x)dx \tag{12}$$

or

$$I_{n+1}^{La}[f] = \frac{(b-a)}{n} \left(\frac{3}{8} f_0 + \frac{7}{6} f_1 + \frac{23}{24} f_2 + \sum_{i=3}^{n-3} f(x_i) + \frac{23}{24} f_{n-2} + \frac{7}{6} f_{n-1} + \frac{3}{8} f_n \right). \tag{13}$$

One of the family of quadrature methods on uniformly spaced points is Derivative-based closed Newton–Cotes numerical quadrature which is derived in [2] and [6]. Derivative-based closed Newton–Cotes-type integration formulas are involved derivative values and function values, therefore there is a number of evaluations of function values and derivative values to solve numerical integral. But in this paper, new closed Composite Newton–Cotes-type integration formula is presented that uses the function values on uniformly spaced intervals only, without any derivative values. To increase the accuracy we divide the given interval into a number of equal subintervals and integrating on each interval by integration rules with abscissas outside Integration Interval. Since there are more parameters for a specified number of intervals, the precision of the new scheme is higher than the standard closed Newton-Cotes formula. These new schemes are presented in the next section.

Theorem 1.1 ([5]). *If Peano kernel does not change its sign on $[a, b]$ then truncation error of $I_{n+1}[f, a, b]$ is*

$$E_{n+1}[f, a, b] = \frac{f^{(n+1)}(\xi)}{(n+1)!} E_{n+1}^{\text{const}}[x^{n+1}, a, b], \tag{14}$$

where $\xi \in (a, b)$ and $E_{n+1}^{\text{const}}[x^{n+1}, a, b] = I[x^{n+1}] - I_{n+1}[x^{n+1}, a, b]$ is called error constant of $I_{n+1}[f, a, b]$. If $E_{n+1}^{\text{const}}[x^{n+1}, a, b]$ also becomes zero, then the error term is obtained for $f(x) = x^{n+2}$.

Theorem 1.2 ([1, 5]). *Let $I_n[f]$ be a fixed n-point rule such that $I_n[1] = b - a$. Let $m \times I_n[f]$ designate the compound rule on $[a, b]$ as defined in (6). Then, if $f(x)$ is a bounded, Riemann-integrable function,*

$$\lim_{m\to\infty} (m \times I_n[f]) = \int_a^b f(x)dx. \tag{15}$$

2. Numerical Integration Methods

Before going to introduce an advanced family of numerical composite integration formulas of closed Newton–Cotes-type that uses the function values on uniformly spaced intervals, we follow the conditions

(1) The number of abscissas in composite n-points closed Newton–Cotes method is equal to the number of abscissas in advanced composite n-points closed Newton–Cotes method.

(2) Length of the each script of (6) is same as length of the each script in new composite closed Newton–Cotes method.

(3) The number of script in (6) is same as in advanced composite closed Newton–Cotes method.

Follow these conditions, we can estimate the area of the first strip in (6) by fitting a $2r+k-2$ degree polynomial through $f_0,\dots,f_{2r+k-2}$ and integrating between x_0 and x_k and similarly for the last strip, estimate the area of the middle strips in (6) by fitting a $2r+k-2$ degree polynomial through $f_{ki-r+1},\dots,f_{ki+k+r-1}$ and integrating between x_{ki} andx_{ki+k}. That is

$$\begin{aligned} I^{k,r}_{n+1}[f] &= \int_{x_0}^{x_k} L_{2r+k-2}(f,x_0,x_{2r+k-2};x)dx + \sum_{i=1}^{(n-2k)/k} \left(\int_{x_{ki}}^{x_{ki+k}} L_{2r+k-2}(f,x_{ki-r+1},x_{ki+k+r-1};x)dx \right) \\ &\quad + \int_{x_{n-k}}^{x_n} L_{2r+k-2}(f,x_{n-2r-k+2},x_n;x)dx, \end{aligned} \tag{16}$$

where k and r $(1 \le r \le k+1)$ are finite numbers and n is multiple of k. This methods are called a family *advanced composite Newton-Cotes methods* of order $2r+k-2$ if k is odd and is of order $2r+k$ if k is even. For $k=1$, $I^{1,2}_{n+1}$ is called advanced trapezoidal method of order 3 which yields better results than the trapezoidal method.

Note. Use $1 \le r \le k+1$ for all k and r, since if $r=3$ then the point x_{i-r+1} does not exist in $[x_{0,},x_n]$ for $i=1$. If $r=1$ then $I^{1,1}_{n+1}$ is same as trapezoidal method.

Now, we will prove convergence of quadrature method (16).

Theorem 2.1. *Let $I^{k,r}_{n+1}[f]$ be an integral formula then for all Riemann integrable functions f which are bounded in $[a,b]$ then*

$$\lim_{n\to\infty} I^{k,r}_{n+1}[f] = I[f]$$

if k and r are finite.

It's clear from Theorem 1.2.

Theorem 2.2. $I^{k,1}_{kn+1}[f] = I^{c}_{kn+1}[f]$ *for* $1 \le k$.

Proof. By substituting r by 1 and n by kn in (16), we get

$$\begin{aligned} I^{k,1}_{kn+1} &= \int_{x_0}^{x_k} L_k(f,x_0,x_k;x)dx + \sum_{i=1}^{(n-2)} \left(\int_{x_{ki}}^{x_{ki+k}} L_k(f,x_{ki},x_{ki+k};x)dx \right) \\ &\quad + \int_{x_{kn-k}}^{x_{kn}} L_k(f,x_{kn-k},x_n;x)dx \\ &= I^{c}_{kn+1}[f]. \end{aligned} \tag{17}$$

Hence the theorem. □

Theorem 2.3. $I^{k,k+1}_{3k+1}[f] = I_{3k+1}[f]$ *for finite.*

Proof. By substituting r by $k+1$ and n by $3k+1$ in (16), we get

$$
\begin{aligned}
I_{3k+1}^{k,k+1}[f] =& \int_{x_0}^{x_k} L_{2(k+1)+k-2}(f,x_0,x_{2(k+1)+k-2};x)dx \\
&+ \sum_{i=1}^{(3k-2k)/k} \left(\int_{x_{ki}}^{x_{ki+k}} L_{2r+k-2}(f,x_{ki-k-1+1},x_{ki+k+k+1-1};x)dx \right) \\
&+ \int_{x_{3k-k}}^{x_{3k}} L_{2(k+1)+k-2}(f,x_{3k-2(k+1)-k+2},x_{3k};x)dx \\
=& \int_{x_0}^{x_k} L_{3k}(f,x_0,x_{3k};x)dx + \sum_{i=1}^{1} \left(\int_{x_{ki}}^{x_{ki+k}} L_{3k}(f,x_{ki-k},x_{ki+2k};x)dx \right) \\
&+ \int_{x_{2k}}^{x_{3k}} L_{3k}(f,x_0,x_{3k};x)dx \\
=& \int_{x_0}^{x_k} L_{3k}(f,x_0,x_{3k};x)dx + \int_{x_k}^{x_{2k}} L_{3k}(f,x_0,x_{3k};x)dx + \int_{x_{2k}}^{x_{3k}} L_{3k}(f,x_0,x_{3k};x)dx \\
=& \int_{x_0}^{x_{3k}} L_{3k}(f,x_0,x_{3k};x)dx \\
=& I_{3k+1}[f].
\end{aligned}
$$

Now, we are going to derive methods.

For $k=1$ and $r=2$ then the integration method is

$$
I_{n+1}^{1,2}[f] = \int_{x_0}^{x_1} L_3(f,x_0,x_3;x)dx + \sum_{i=1}^{n-2} \left(\int_{x_i}^{x_{i+1}} L_3(f,x_{i-1},x_{i+2};x)dx \right) + \int_{x_{n-1}}^{x_n} L_3(f,x_{n-3},x_n;x)dx. \tag{18}
$$

The formulas of $\int_{x_0}^{x_1} L_3(f,x_0,x_3;x)dx$, $\int_{x_{n-1}}^{x_n} L_3(f,x_{n-3},x_n;x)dx$ and $\int_{x_i}^{x_{i+1}} L_3(f,x_{i-1},x_{i+2};x)dx$ are obtained using the method of undetermined coefficients which are based on the concept of the precision of the quadrature formula. So, we get

$$
\int_{x_0}^{x_1} L_3(f,x_0,x_3;x)dx = \frac{3h}{8}f_0 + \frac{19h}{24}f_1 - \frac{5h}{24}f_2 + \frac{h}{24}f_3 - \frac{19}{30.4!}h^5 f^{(4)}(\xi_0), \quad \xi_0 \in (a,b), \tag{19}
$$

$$
\int_{x_{n-1}}^{x_n} L_3(f,x_{n-3},x_n;x)dx = \frac{h}{24}f_{n-3} - \frac{5h}{24}f_{n-2} + \frac{19h}{24}f_{n-1} + \frac{3h}{8}f_n - \frac{19}{30.4!}h^5 f^{(4)}(\xi_{n-1}), \quad \xi_{n-1} \in (a,b) \tag{20}
$$

and

$$
\int_{x_i}^{x_{i+1}} L_3(f,x_{i-1},x_{i+2};x)dx = -\frac{h}{24}f_{i-1} + \frac{13h}{24}f_i + \frac{13h}{24}f_{i+1} - \frac{h}{24}f_{i+2} + \frac{11}{30.4!}h^5 f^{(4)}(\xi_i), \quad \xi_i \in (a,b). \tag{21}
$$

Plug these values in (18), we get

$$
\begin{aligned}
I_{n+1}^{1,2}[f] =& \frac{h}{3}\left(f_0 + \frac{31}{8}f_1 + \frac{5}{2}f_2 + \frac{25}{8}f_3 + \frac{25}{8}f_{n-3} + \frac{5}{2}f_{n-2} + \frac{31}{8}f_{n-1} + f_n \right) \\
&+ h\sum_{i=4}^{n-4} f_i + \left(\frac{11n-60}{30.4!}\right)h^5 f^{(4)}(\xi),
\end{aligned} \tag{22}
$$

where $\xi \in (a,b)$ and this method is called an *advanced trapezoidal method* of the order of accuracy 3, which yields better results than the trapezoidal method and it is a positive method. In this method n is not less than 3 and any finite number. This formula gives equal weights to all ordinates except the first four and last four.

Take $k=2$ and $r=2$ from (16) then the integration method is

$$I_{n+1}^{2,2}[f]=\int_{x_0}^{x_2} L_4(f,x_0,x_4;x)dx+\sum_{i=1}^{(n-4)/2}\left(\int_{x_{2i}}^{x_{2i+2}} L_4(f,x_{2i-1},x_{2i+3};x)dx\right)+\int_{x_{n-2}}^{x_n} L_4(f,x_{n-4},x_n;x)dx \tag{23}$$

But we know

$$\int_{x_0}^{x_2} L_3(f,x_0,x_4;x)dx=\frac{29h}{90}f_0+\frac{62h}{45}f_1+\frac{4h}{15}f_2+\frac{2h}{45}f_3-\frac{h}{90}f_4+\frac{4}{3.5!}h^6f^{(5)}(\xi_0),\quad \xi_0\in(a,b),$$

$$\int_{x_{n-2}}^{x_n} L_3(f,x_{n-4},x_n;x)dx=-\frac{h}{90}f_{n-4}+\frac{2h}{45}f_{n-3}+\frac{4h}{15}f_{n-2}+\frac{62h}{45}f_{n-1}+\frac{29h}{90}f_n-\frac{4}{3.5!}h^6f^{(5)}(\xi_{n-2}),$$

and

$$\begin{aligned}\int_{x_{2i}}^{x_{2i+2}} L_4(f,x_{2i-1},x_{2i+3};x)dx=&-\frac{h}{90}f_{2i-1}+\frac{17h}{45}f_{2i}+\frac{19h}{15}f_{2i+1}+\frac{17h}{45}f_{2i+2}\\&-\frac{h}{90}f_{2i+3}+\frac{20}{21.5!}h^7f^{(6)}(\xi_{2i}),\end{aligned}$$

where $\xi_{n-2}\in(a,b)$ and $\xi_{2i}\in(a,b)$. Here $\int_{x_0}^{x_2} L_3(f,x_0,x_4;x)dx$ and $\int_{x_{n-2}}^{x_n} L_3(f,x_{n-4},x_n;x)dx$ are integration methods of order 5 whose error constants are equal in magnitude but differ in sign, then we have $M=\int_{x_0}^{x_2} L_3(f,x_0,x_4;x)dx+\int_{x_{n-2}}^{x_n} L_3(f,x_{n-4},x_n;x)dx$ is integration method of order 6. Therefore, this method of M is exact for polynomials upto degree 6. The error of M is $-\frac{128h^7}{21.6!}f^{(7)}(\xi),\xi\in(x_0,x_n)$.

Substituting these values in (23), we get

$$\begin{aligned}I_{n+1}^{2,2}[f]=&\frac{h}{90}(29(f_0+f_n)+123(f_1+f_{n-1})+58(f_2+f_{n-2})+117(f_3+f_{n-3})+67(f_4+f_{n-4}))\\&+\frac{h}{45}\left(56\sum_{i=2}^{(n-6)/2}f_{2i+1}+39\sum_{i=2}^{(n-8)/2}f_{2i+2}\right)+-\frac{(20n+296)}{42.6!}h^7f^{(6)}(\xi),\end{aligned} \tag{24}$$

where $\xi\in(x_0,x_n)$, this method is called *Advanced Simpson's method* of order os accuracy 6 and in this method n is even and not less then 4.

For $k=2$and $r=3$ then the integration method is

$$\begin{aligned}I_{n+1}^{2,3}[f]=&\frac{h}{3780}\Big(1144(f_0+f_n)+5568(f_1+f_{n-1})+1541(f_2+f_{n-2})+5944(f_3+f_{n-3})\\&+2204(f_4+f_{n-4})+4808(f_5+f_{n-5})+4808(f_5+f_{n-5})+2979(f_6+f_{n-6})\\&+4544\sum_{i=7}^{n/2}f_{2i-7}+3016\sum_{i=8}^{n/2}f_{2i-8}\Big)-\frac{5648}{45.8!}(n-2)h^9f^{(8)}(\xi),\end{aligned} \tag{25}$$

where $\xi\in(x_0,x_n)$, this method is called 2-*Advanced Simpson's method* and in this method n is even and not less then 6.

3. Error Estimate

Some important notes of error are given bellow by without any proof.

Note. If Peano kernel does not change its sign on $[a,b]$ and $f \in C^{2r+k}[a,b]$ then from Theorem 1.1

(1) Truncation error of $\int_{x_0}^{x_k} L_{2r+k-2}(f,x_0,x_{2r+k-2};x)dx$ is

$$E_{2r+k-1}[f,x_0,x_k] = \begin{cases} \dfrac{f^{(2r+k-1)}(\xi)}{(2r+k-1)!} E^{c_0}[x^{2r+k-1}] & \text{if } k \text{ is odd,} \\ \dfrac{f^{(2r+k)}(\xi)}{(2r+k)!} E^{C_0}[x^{2r+k}] & \text{if } k \text{ is even.} \end{cases} \tag{26}$$

(2) Truncation error of $\int_{x_{ki}}^{x_{ki+k}} L_{2r+k-2}(f,x_{ki-r+1},x_{ki+k+r-1};x)dx$ is

$$E_{2r+k-1}[f,x_{ki-r+1},x_{ki+k+r-1}] = \begin{cases} \dfrac{f^{(2r+k-1)}(\xi)}{(2r+k-1)!} E^{c_i}[x^{2r+k-1}] & \text{if } k \text{ is odd,} \\ \dfrac{f^{(2r+k+1)}(\xi)}{(2r+k+1)!} E^{C_i}[x^{2r+k+1}] & \text{if } k \text{ is even.} \end{cases} \tag{27}$$

(3) The error constant

$$E^{c_0}[x^{2r+k-1}] = E^{c_{n=k}}[x^{2r+k-1}] \quad \text{if } k \text{ is odd} \tag{28}$$

and

$$E^{C_0}[x^{2r+k}] = -E^{C_{n-k}}[x^{2r+k}] \quad \text{if } k \text{ is even.} \tag{29}$$

Theorem 3.1. *If Peano kernel does not change its sign on $[a,b]$, $f \in C^{2r+k+1}[a,b]$ and $I_{n+1}^{k,r}[f]$ is composite quadrature method on uniformly spaced points with space h then the error of $I_{n+1}^{k,r}[f]$ is*

$$E_{n+1}^{k,r}[f,a,b] \le \begin{cases} c(n,k,r)h^{2r+k} \dfrac{f^{(2r+k-1)}(\xi)}{(2r+k-1)!} & \text{if } k \text{ is odd} \\ C(n,k,r)h^{2r+k+2} \dfrac{f^{(2r+k+1)}(\zeta)}{(2r+k+1)!} & \text{if } k \text{ is even} \end{cases}$$

where $\xi,\zeta \in [a,b]$ and the constants $c(n,k,r)$ and $C(n,k,r)$ may depends on $I_{n+1}^{k,r}[f]$ but are independent of a, b, and f.

Proof. The error terms are found using the concept of precision. We have Peano kernel does not change its sign on $[a,b]$ and $f \in C^{2r+k+1}[a,b]$.

We have from (26) and (27), the error of (16) is

$$E_{n+1}^{k,r}[f,a,b] = E_{2r+k-1}[f,x_0,x_k] + \sum_{i=1}^{(n-2k)/k} E_{2r+k-1}[f,x_{ki-r+1},x_{ki+k+r-1}] + E_{2r+k-1}[f,x_{n-k},x_n],$$

$$E_{n+1}^{k,r}[f,a,b] = \begin{cases} \dfrac{f^{(2r+k-1)}(\xi_0)}{(2r+k-1)!}E^{c_o}[x^{2r+k-1}] + \displaystyle\sum_{i=1}^{(n-2k)/k} \dfrac{f^{(2r+k-1)}(\xi_i)}{(2r+k-1)!}E^{c_i}[x^{2r+k-1}] & \text{if } k \text{ is odd} \\ + \dfrac{f^{(2r+k-1)}(\xi_{n-k})}{(2r+k-1)!}E^{c_{n-k}}[x^{2r+k-1}] & \\ \dfrac{f^{(2r+k)}(\zeta_0)}{(2r+k)!}E^{C_0}[x^{2r+k}] + \dfrac{f^{(2r+k)}(\zeta_{n-k})}{(2r+k)!}E^{C_0}E^{C_n}[x^{2r+k}] & \text{if } k \text{ is even} \end{cases}$$

where $\xi_i, \zeta_i \in (x_{ki}, x_{ki+k})$, $i = 0, 1, \dots, n-k$.

If k is even, from (29) the error constant of $N = \int_{x_0}^{x_k} L_{2r+k-2}(f, x_0, x_{2r+k-2}; x)dx + \int_{x_{n-k}}^{x_n} L_{2r+k-2}(f, x_{n-2r-k+2}, x_n; x)dx$ is zero. This mean this quadrature method is exact for upto polynomial of degree $2r+k$, then the error constant of N is $E^{C_0}[x^{2r+k+1}] + +E^{C_n}[x^{2r+k+1}]$. Take

$$\xi = \max_{a \le \xi_i \le b} \{f^{(2r+k-1)}(\xi_i),\ i = 0, \dots, n-k\}$$

and

$$\zeta = \max_{a \le \zeta_i \le b} \{f^{(2r+k+1)}(\zeta_i),\ i = 0, \dots, n-k\}.$$

From (28) and (29), we get

$$E_{n+1}^{k,r}[f,a,b] = \begin{cases} \dfrac{f^{(2r+k-1)}(\xi)}{(2r+k-1)!}\left(2E^{c_o}[x^{2r+k-1}] + \dfrac{n-2k}{k}E^{c_i}[x^{2r+k-1}]\right) & \text{if } k \text{ is odd} \\ \dfrac{f^{(2r+k+1)}(\zeta)}{(2r+k+1)!}\left(E^{C_0}[x^{2r+k+1}] + E^{C_n}[x^{2r+k+1}] + \dfrac{n-2k}{k}E^{C_i}[x^{2r+k+1}]\right) & \text{if } k \text{ is even} \end{cases}$$

$$E_{n+1}^{k,r}[f,a,b] = \begin{cases} c(n,k,r)h^{2r+k}\dfrac{f^{(2r+k-1)}(\xi)}{(2r+k-1)!} & \text{if } k \text{ is odd} \\ C(n,k,r)h^{2r+k+2}\dfrac{f^{(2r+k+1)}(\zeta)}{(2r+k+1)!} & \text{if } k \text{ is even} \end{cases}$$

where

$$c(n,k,r) = 2E^{c_o}[x^{2r+k-1}] + \frac{n-2k}{k}E^{c_i}[x^{2r+k-1}]$$

and

$$C(n,k,r) = E^{C_0}[x^{2r+k+1}] + E^{C_n}[x^{2r+k+1}] + \frac{n-2k}{k}E^{C_i}[x^{2r+k+1}].$$

Hence the theorem. □

4. Test Results

We have used functions that are analytically integrable to test the methods. For the same definite integral, we have used an increasing number of points, and 6 methods at each number of points. When the results are very large or very small,therefore we used the absolute error. The value is E given by the formula

$$E[f] = |I_{n+1}[f] - I[f]|$$

where $I_{n+1}[f]$ is the estimate, $I[f]$ is the exact value, and $E[f]$ is the absolutely error. $E[f]$ will be a measure of number of accurate digits, so $E = -16$ is the best we can attain using MATLAB. There is no few absolute errors of Boole's method for n like $n = 10, 50$ and 30. Since n is multiple of 4 in Boole's method. We applied these methods to

Problem 1. $\int_0^{\pi/2} \frac{1}{1+\cos(x)} dx = 1.$

Problem 2. $\int_0^{\pi/2} \cos^3(x) dx = \frac{2}{3}.$

Problem 3. $\int_0^{\pi/2} \frac{1}{1+x} dx = -\ln(2) + \ln(\pi+2) = 0.94421570569605539178.$

Problem 4. $\int_0^{\pi/2} \sqrt{x}\, dx = \left(\frac{1}{6}\right) * \sqrt{2} * \pi^{3/2} = 1.3124674954768683121.$

Problem 5. $\int_1^3 \frac{e^x}{x} dx = 8.0387147542694798025.$

We get Table 1.

Table 1. Absolutely errors and computational cost for each of these methods was analyzed and compared to the standard closed composite Newton-Cotes formula.

Integral	n	Trapezoidal	Simpson	Boole's	Advanced methods		
					Trapezoidal	Simpsons	2-Simpson
Problem 1	10	0.002052	1.6338e-05	–	6.7607e-06	3.5064e-06	5.8674e-07
	20	0.00051378	1.0476e-06	2.8264e-08	1.3412e-06	8.8264e-08	5.8589e-09
	50	8.224e-05	2.7019e-08	–	5.564e-08	4.9228e-10	7.1483e-12
	100	2.0561e-05	1.6905e-09	1.9336e-12	4.0302e-09	8.5691e-12	3.4861e-14
	1000	2.0562e-07	1.6920e-13	4.4409e-16	4.5852e-13	-1.4433e-15	2.2204e-16
Problem 2	10	5.1034e-06	2.0904e-05	–	9.2599e-05	1.295e-05	2.0941e-06
	20	3.1755e-07	1.2777e-06	3.0705e-08	5.425e-06	1.9007e-07	9.3247e-09
	50	8.1193e-09	3.2508e-08	–	1.1348e-07	6.8674e-10	5.3803e-12
	100	5.0737e-10	2.03e-09	1.9098e-12	6.3782e-09	1.0107e-11	1.8874e-14
	1000	5.0959e-14	2.0328e-13	2.2204e-16	5.6655e-13	0	0
Problem 3	10	0.0017402	1.8752e-05	–	3.8455e-06	5.2787e-06	1.214e-06
	20	0.00043595	1.2212e-06	5.2528e-08	1.2923e-06	1.4791e-07	1.4939e-08
	50	6.9794e-05	3.1651e-08	–	6.0695e-08	8.9933e-10	2.2024e-11
	100	1.745e-05	1.9817e-09	3.7814e-12	4.5597e-09	1.6193e-11	1.1668e-13
	1000	1.7451e-07	1.9917e-13	0	5.3479e-13	4.4409e-16	2.2204e-16
Problem 4	10	0.012122	0.0050538	–	0.0049674	0.0047239	0.0042295
	50	0.0011248	0.00045206	–	0.00044409	0.00042253	0.00037831
	100	-0.00040106	0.00015983	-0.00014035	0.00015701	0.00014939	0.00013375
	1000	1.286e-05	5.0542e-06	-4.4381e-06	4.9651e-06	4.7241e-06	4.2297e-06
	10000	4.0845e-07	1.5983e-07	-1.4035e-07	1.5701e-07	1.4939e-07	1.3375e-07
Problem 5	20	0.0037184	4.5962e-06	2.1398e-07	-5.1938e-06	5.5329e-07	7.4436e-08
	30	0.0016529	9.1621e-07	–	-1.404e-06	6.2718e-08	4.8994e-09
	40	0.00092981	2.9085e-07	3.8227e-09	-5.1637e-07	1.2778e-08	6.5033e-10
	50	0.0005951	1.1931e-07	–	-2.3122e-07	3.6425e-09	1.3022e-10
	100	0.00014878	7.4724e-09	1.635e-11	-1.7243e-08	6.7741e-11	7.3719e-13

From Table 1, we are observing the following below results.

(1) Advanced Trapezoidal yields better results than the trapezoidal method.

(2) Advanced Trapezoidal yields good or approximate equal results than the Simpson's method.

(3) Advanced Simpson's gives the better results of trapezoidal, Simpson and Advanced trapezoidal methods. Also, it gaves approximate equal results than the Boole's method.

(4) And also 2-Advanced Simpson's gives the best results of trapezoidal, Simpson's, Boole's, Advanced trapezoidal, Advanced Simpson's methods and Boole's method.

We have Durand and Lacriox methods which yields better results than the trapezoidal method. Now comparing these methods to Advanced trapezoidal method. Table 2 shows absolute errors of Problem 1, Problem 3 and Problem 5 to different n values.

Table 2. Comparing absolute errors of Advanced trapezoidal method to Durand and Lacriox methods.

Integral	n	Durand method	Lacriox methos	Advanced Trapezoidal
problem 1	10	0.00032179	5.4576e-05	6.7607e-06
	20	4.2675e-05	4.1172e-06	1.3412e-06
	50	2.8341e-06	1.1855e-07	5.564e-08
	100	3.5876e-07	7.7129e-09	4.0302e-09
	1000	3.6289e-10	7.9958e-13	4.5852e-13
problem 3	10	0.00029448	5.9011e-05	3.8455e-06
	20	3.9575e-05	4.6133e-06	1.2923e-06
	50	2.6512e-06	1.3631e-07	6.0695e-08
	100	3.3662e-07	8.9533e-09	4.5597e-09
	1000	3.4145e-10	9.388e-13	5.3479e-13
Problem 5	10	0.0019663	0.00022718	2.3361e-05
	20	0.00025626	1.7513e-05	5.1938e-06
	50	1.6848e-05	5.1477e-07	2.3122e-07
	100	2.1257e-06	3.3778e-08	1.7243e-08
	1000	2.1439e-09	3.5332e-12	2.0197e-12

From Table 2, we observe, the advanced trapezoidal method gives the best results of Durand and Lacriox methods.

5. Conclusion

A new family of the closed composite Newton-Cotes type of quadrature formula was presented, that include the use of function values which are outside of interval in each script, to increase the order of accuracy of the numerical approximations of definite integrals. Using the concept of precision, where the quadrature formula exactly integrates monomials up to a certain degree, a system of linear equations for the weights was created and solved. In general, the order of the quadrature rule is $2r+k-2$ if k is odd and is of order $2r+k$ if k is even.

The computational cost for each of these methods was analyzed and compared to the standard closed composite Newton-Cotes formula for five different integrals $\int_0^{\pi/2} \frac{1}{1+\cos(x)}\,dx$,

$\int_0^{\pi/2} \cos^3(x)\,dx$, $\int_0^{\pi/2} \frac{1}{1+x}\,dx$, $\int_0^{\pi/2} \sqrt{x}\,dx$ and $\int_1^3 \frac{e^x}{x}\,dx$. The new composite quadrature formula were superior computationally to the same order closed Newton–Cotes formula.

One of the closed composite Newton-Cotes formulae is Simpson's method (order of accuracy 3), one of the new closed composite Newton-Cotes formulae is an advanced trapezoidal method (order of accuracy 3). these two methods are the same order of accuracy 3 and give approximately same values. In Simpson's method n must be an even number and minimum number is 2, but in advanced trapezoidal method n is any number and minimum number is 3, in computationally n will take lodge number, so the minimum number of n is not considered in computational. similarly, Boole's method and advanced Simpson's method are the same order of accuracy if $n = 10, 50$ then Boole's method not applicable, since in Boole's method n must be multiple of 4, but advanced Simpson's method is applicable for an even number. This is one of the advantage of new family of closed composite Newton-Cotes formula. We have developed two advanced Simpson's methods which are applicable to any big even number.

The error bounds for the composite quadrature formula were originally derived by the author using the concept of precision, by making certain unverifiable assumptions about the higher order terms. Mentioned the related MATLAB [4] codes in Appendix A.

Appendix A. Few MATLAB Codes for Advanced Closed Newton–Cotes Numerical Quadrature

The MATLAB codes for the composite quadrature formula were originally derived by the authors.

A1. MATLAB Codes for Advanced Trapezoidal Method

```
function e = adtr(f,a,b,n)
% n is any number not less then 3
h = (b-a)/n;
p = 0;
i = 1:(n+1);
x(i) = a + (i-1)*h;
for i = 1:(n-2);
p = p+(-f(x(i))+13*f(x(i+1))+13*f(x(i+2))-f(x(i+3)));
end
I(3) =(1/24)*h*p;
I(1)=(h/24)*(9*f(x(1))+19*f(x(2))-5*f(x(3))+f(x(4)));
I(2)=(h/24)*(19*f(x(n))+9*f(x(n+1))-5*f(x(n-1))+f(x(n-2)));
x=I(1)+I(2)+I(3);
e=x-integral(f,a,b);
end
```

A2. MATLAB Codes for Advanced Simpson's method

```
function e = adsim(f,a,b,n)
% n must be multiple of 2 and not less then 4
h = (b-a)/n;
p = 0;
i = 1:(n+1);
x(i) = a + (i-1)*h;
for i = 2:2:(n-3);
p = p+(-f(x(i))+34*f(x(i+1))+114*f(x(i+2))+34*f(x(i+3))-f(x(i+4)));
end
I(3) =(1/90)*h*p;
I(1)=(h/90)*(29*f(x(1))+124*f(x(2))+24*f(x(3))+4*f(x(4))-f(x(5)));
I(2)=(h/90)*(4*f(x(n-2))-f(x(n-3))+24*f(x(n-1))+124*f(x(n))+29*f(x(n+1)));
x=I(3)+I(1)+I(2);
e=x-integral(f,a,b);
end
```

A3. MATLAB Codes for Second Advanced Simpson's method

```
function e = p2p7m(f,a,b,n)
% n must be multiple of 2 and not less then 6
h = (b-a)/n;
p = 0;
i = 1:(n+1);
x(i) = a + (i-1)*h;
for i = 2:2:(n-4);
p = p+((1/756)*f(x(i-1))-(2/105)*f(x(i))+(167/420)*f(x(i+1))+(1172/945)*f(x(i+2))+(167/420)*f(x(i+3))-(2/105)*f(x(i+4))+(1/756)*f(x(i+5)));
end
I(3) =h*p;
I(1)=h*((1139/3780)*f(x(1))+(94/63)*f(x(2))+(11/1260)*f(x(3))+(332/945)*f(x(4))-(269/1260)*f(x(5))+(22/315)*f(x(6))-(37/3780)*f(x(7)));
I(2)=h*((1139/3780)*f(x(n+1))+(94/63)*f(x(n))+(11/1260)*f(x(n-1))+(332/945)*f(x(n-2))-(269/1260)*f(x(n-3))+(22/315)*f(x(n-4))-
(37/3780)*f(x(n-5)));
x=I(3)+I(1)+I(2);
e=x-integral(f,a,b);
end
```

Acknowledgement

We gratefully acknowledge the support of the UGC (University Grants Commission) JRF and NET Fellowship, India.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] H. Brass and K. Petras, *Quadrature Theory: The Theory of Numerical Integration on a Compact Interval*, No. 178, American Mathematical Soc. (2011).

[2] C.O.E. Burg, Derivative-based closed Newton–Cotes numerical quadrature, *Applied Mathematics and Computation* **218** (2012), 7052 – 7065, DOI: 10.1016/j.amc.2011.12.060.

[3] M. Chalpuri and J. Sucharitha, Open-type quadrature methods with equispaced nodes and a maximal polynomial degree of exactness, *International Journal of Mathematics and Its Applications* **55** (2017), 7.

[4] S.C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, McGraw-Hill Science (2004).

[5] P.J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Courier Corporation (2007).

[6] S.M. Hashemiparast, M.R. Eslahchi and M. Dehghan, Numerical integration using the derivatives, *Applied Mathematics and Computation* **180**(1) (2006), 194 – 199, DOI: 10.1016/j.amc.2005.11.169.

[7] T.M. Mills and S.J. Smith, A note on the Newton-Cotes integration formula, *Journal of Approximation Theory* **66**(1) (1991), 98 – 105, DOI: 10.1016/0021-9045(91)90059-J.

[8] W.W. Piegorsch, Durand's rules for approximate integration, *Historia Mathematica* **16**(4) (1989), 324 – 333, DOI: 10.1016/0315-0860(89)90079-7.

[9] E. Sermutlu, Comparison of Newton–Cotes and Gaussian methods of quadrature, *Applied Mathematics and Computation* **171** (2005), 1048 – 1057, DOI: 10.1016/j.amc.2005.01.102.