

Fast Numerical Simpson and Boole Integration by Using the Derivatives at the Boundaries

Michael Engler*

Department of Business Administration and Engineering, Furtwangen University, Furtwangen, Germany

Research Article

Received: 17-Nov-2023, Manuscript No. JSMS- 23-120358; **Editor assigned:** 20-Nov-2023, Pre QC No. JSMS- 23-120358 (PQ); **Reviewed:** 04-Dec-2023, QC No. JSMS- 23-120358; **Revised:** 11-Dec-2023, Manuscript No. JSMS- 23-120358 (R); **Published:** 18-Dec-2023, DOI: 10.4172/RRJ Stats Math Sci. 9.4.001

***For Correspondence:** Michael Engler, Department of Business Administration and Engineering, Furtwangen University, Furtwangen, Germany

E-mail: michael.engler@hs-furtwangen.de

Citation: Engler M. Fast Numerical Simpson and Boole Integration by Using the Derivatives at the Boundaries. RRJ Stats Math Sci.2023.9.001.

Copyright: © 2023 Engler M. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

This article shows the extension of the closed Newton-Cotes numerical integration of Simpson's and Boole's rule by using the odd derivatives of the function at the boundaries of the integration interval. The derivatives can be used to efficiently increase the convergence order of numerical integration and a fast decrease of the error. Furthermore, due to its simplicity, it is very easy to write into program code, which is also shown. The error estimation is given and proven. Also, the method is confirmed with two different examples for numerical integration, of π and of the integral of the Gaussian distribution. Here, the method is compared to some common numerical integration methods, showing comparably faster convergence.

Keywords: Numerical integration; Numerical quadrature; Derivative-based quadrature; Closed newton-cotes integration; Quadrature error estimation

MSC Classification: 65D30, 65D32

INTRODUCTION

Given a function $f : [a, b] \rightarrow \mathbb{R}$. Supposed one wants to calculate the integral $I := \int_a^b f(x) dx$, and a primitive of f is not known, does not exist in closed form or is too difficult to be calculated. Then, numerical calculation methods are needed, one of which is presented here. A very good comprehension of the different methods is given by Davis and Rabinowitz [1]. For numerical integration as presented here, the interval $[a, b]$ is dissected into n equidistant intervals of width h :

$$h := \frac{b - a}{n}. \tag{1}$$

There are $n + 1$ x -values, denominated $a = x_0, x_1, \dots, x_{n-1}, x_n = b$. Within each interval, f is approximated by an interpolation function which is then integrated instead of f . The results of each approximated intervals are summed over all intervals and give an approximate solution of the integral. When $h \rightarrow 0$, the numerical solution approaches the real value of the integral. One of the simplest numerical integration schemes is the Trapezoidal Quadrature. Here, the value I_T of the numerical integration is calculated to:

$$I_T := \frac{h}{2} \sum_{i=0}^{n-1} f(x_i) + f(x_i + h), \tag{2}$$

with the error E_T being of the order $h^2 (||f''||_{max} := \max_{x \in [a,b]} |f''(x)|)$:

$$E_T := |I - I_T| \leq \frac{1}{12} \cdot h^2 \cdot (b - a) \cdot ||f''||_{max}.$$

When one uses not only the function values, but also the values of its derivatives, the convergence order and speed strongly improves. It is already known that the use of the first derivative at the boundaries a and b lead to an increase of convergence order by h^2 [1]. The method is known as the Euler-MacLaurin summation formula [2]. This approach is also shown in different characteristics from Ujevic and Catinas et al [3,4]. Davis, Rabinowitz and Stancu and Stroud have shown that the integral of the function f can be approximated by the function values and the values of arbitrary derivatives of the function [5]. Here, the derivatives at all nodes are used for the calculation. This method has been detailed by Burg and Burg and Degny [6,7]. They give the error estimations for different Newton-Cotes quadrature rules in combination with the first, second and third derivative and they show that both the convergence order increases and the error decreases in these cases.

In this article, we show that when using Newton-Cotes Quadrature in combination with any odd derivative of f at the boundaries, the convergence order vastly increases and the error decreases with each derivative being used. One great advantage of the odd derivatives is that only the derivatives at the boundaries are needed. We give the Simpson Quadrature formula extended with any odd derivative and its error estimation, showing that the error is of the order h^{2m+4} , where m is the highest number of the odd derivative being used ($m=1$, first derivative, $m=2$, third derivative, and so on). Furthermore, the error coefficient decreases vastly with every higher derivative being used. The same holds for the extended Boole Quadrature formula, where the error is of the order h^{2m+6} . Both algorithms, which are very easy to implement, are given.

METHODOLOGY

Simpson Quadrature with Odd Derivatives

The Trapezoidal Quadrature can be extended by using not only two function values per interval, but by interpolating multiple intervals with values of different weights. These methods lead to the well-known Newton-Cotes formulas, e.g. Simpson’s rule and Boole’s rule. Details are shown in any textbook of basic numerical mathematics, e.g. Simpson’s rule is given by using two combined intervals and three values [2]:

$$I_S := h \sum_{i=0}^{\frac{n}{2}-1} w_1 f(x_{2i}) + w_2 f(x_{2i} + h) + w_3 f(x_{2i} + 2h), \tag{3}$$

with the weights $w_1 = w_3 = \frac{1}{3}$ and $w_2 = \frac{4}{3}$. The error of Simpson’s rule is converging by the order of h^4 , two orders of magnitude faster than that of the Trapezoidal rule. The error coefficient is decreased by a factor of 15 (1/180 compared to 1/12):

$$E_S := |I - I_S| \leq \frac{1}{180} \cdot h^4 \cdot (b - a) \cdot \|f^{(4)}\|_{max}.$$

Definition of Simpson Odd Derivative

We will now extend the classical Simpson rule by the odd derivatives at the borders, giving the so-called "Simpson Odd Derivative Quadrature SOD" in Table 1.

Table 1. Coefficients of the Simpson Odd Derivative Quadrature up to $m=5$ (9th derivative).

m	$w_1=w_3$	w_2	α_1	α_2	α_3	α_4	α_5	Error coef. γ
1	$\frac{7}{15}$	$\frac{16}{15}$	$\frac{1}{15}$					$\frac{1}{4725}$
2	$\frac{31}{63}$	$\frac{64}{63}$	$\frac{5}{63}$	$-\frac{1}{945}$				$\frac{1}{198450}$
3	$\frac{127}{255}$	$\frac{256}{255}$	$\frac{7}{85}$	$-\frac{1}{765}$	$\frac{2}{80325}$			$\frac{1}{7952175}$
4	$\frac{511}{1023}$	$\frac{1024}{1023}$	$\frac{85}{1023}$	$-\frac{7}{5115}$	$\frac{2}{64449}$	$-\frac{1}{1611225}$		$\frac{3}{945294749}$
5	$\frac{2047}{4095}$	$\frac{4096}{4095}$	$\frac{341}{4095}$	$-\frac{17}{12285}$	$\frac{2}{61425}$	$-\frac{1}{1289925}$	$\frac{1}{63851288}$	$\frac{2}{24902002331}$

Definition 1 (Simpson Odd Derivative Quadrature SOD): Given the integral $I := \int_a^b f(x) dx$ and the Simpson Rule I_S defined by Equation 3 with n even and equidistant dissections of the interval of width h (defined by Equation 1) of the function $f \in C^{2m+4} [a, b]$, with a natural number $m \geq 1$. Given also the odd derivatives of f at the borders with $|f^{(2j-1)}(a)| < \infty$ and $|f^{(2j-1)}(b)| < \infty$ for $j = 1, \dots, m$ with at least one $f^{(2j-1)}(a)$ or $f^{(2j-1)}(b) \neq 0$.

A fast converging formula for the numerical value of the integral is then calculated by:

$$I_{SOD} := I_S + \sum_{j=1}^m \alpha_j \cdot h^{2j} \cdot (f^{(2j-1)}(a) - f^{(2j-1)}(b)). \tag{4}$$

The weights w_1 differ compared to Equation 3 with the number of derivatives m being used. They are shown together with the α_j in Table 1. We will now calculate the coefficients given in Table 1. For this, we will approximate f by its Taylor series up to the degree $2m + 3$ within the interval i ranging from x_i to $x_i + 2h$. Then follows:

$$\int_{x_i}^{x_i+2h} f(x) dx \approx \sum_{j=0}^{2m+3} \frac{(2h)^{j+1}}{(j+1)!} f^{(j)}(x_i). \tag{5}$$

Using the definition of I_{SOD} , Equation 4, and considering only the interval from x_i to $x_i + 2h$, the Taylor approximation up to the degree $2m + 3$ gives:

$$\begin{aligned} I_{SOD,i} &\approx h(w_1 f(x_i) + w_2 f(x_i + h) + w_3 f(x_i + 2h)) + \\ &\quad + \sum_{j=1}^m \alpha_j h^{2j} (f^{(2j-1)}(x_i) - f^{(2j-1)}(x_i + 2h)) \\ &= w_1 h f(x_i) + w_2 \sum_{j=0}^{2m+3} \frac{h^{j+1}}{j!} f^{(j)}(x_i) + w_3 \sum_{j=0}^{2m+3} \frac{2^j h^{j+1}}{j!} f^{(j)}(x_i) - \\ &\quad - \alpha_1 \sum_{j=2}^{2m+3} \frac{2^{j-1} h^{j+1}}{(j-1)!} f^{(j)}(x_i) - \dots - \alpha_m \sum_{j=2m}^{2m+3} \frac{2^{j-2m+1} h^{j+1}}{(j-2m+1)!} f^{(j)}(x_i). \end{aligned} \tag{6}$$

Adding all $n/2$ intervals $I_{SOD,i}$ together gives the approximation of I_{SOD} . Note that only the leftmost and rightmost derivatives remain, as the inner derivatives cancel each other out. This works only when the odd derivatives are used, when even derivatives are used, the inner derivative values remain and the method will become much less efficient. For the calculation of the optimal w_1, w_2, w_3 and α_j , Equation 5 and Equation 6 have to be set equal, as only then the approximation SOD gives the integral approximation. Comparing the coefficients with the corresponding derivatives, the following LSE with $4 + 2m$ rows and $3 + m$ columns results:

$$\begin{pmatrix} 1 & 1 & 2^0 & 0 & \dots & 0 \\ 0 & 1 & 2^1 & 0 & \dots & 0 \\ 0 & 1 & 2^2 & -\frac{2!}{1!} \cdot 2^1 & \dots & 0 \\ 0 & 1 & 2^3 & -\frac{3!}{2!} \cdot 2^2 & \dots & 0 \\ & & & \vdots & & \\ 0 & 1 & 2^{2m} & -\frac{(2m)!}{(2m-1)!} \cdot 2^{2m-1} & \dots & -\frac{(2m)!}{(1)!} \cdot 2^1 \\ & & & \vdots & & \\ 0 & 1 & 2^{2m+3} & -\frac{(2m+3)!}{(2m+2)!} \cdot 2^{2m+2} & \dots & -\frac{(2m+3)!}{(4)!} \cdot 2^4 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} 2^1 \\ 1 \\ \vdots \\ 2^{2m+4} \\ 2^{m+4} \end{pmatrix} \tag{7}$$

Although the LSE has more rows than variables, the solution is unique (without proof) and gives the coefficients $w_1, w_2, w_3,$ and $\alpha_j,$ for the Simpson-Odd-Derivative Quadrature formula, Equation 4. The values when using the first five odd derivatives ($m = 5$) have been and are shown in Table 1. Higher derivatives can be calculated from Equation 7, but the calculation becomes more awkward due to the small numbers where one has to take care about rounding errors.

Please note again, that only the odd derivatives at the borders of the interval are needed and with each odd order being added, the convergence is by order of 2 faster. Additionally, the error coefficient γ decreases strongly, as will be shown below. Please note that in the case of $f^{(2j-1)}(a) = f^{(2j-1)}(b)$ for $j \in 1, \dots, m,$ the method is still applicable

but not so effective. Overall, the method is applicable when $|f(a)|$ and $|f(b)|$ are $< \infty$ and it becomes more efficient when the higher derivatives of f are not becoming too large with higher orders.

The quadrature I_{SOD} is “optimal” in the sense that the approximation is perfect for any polynomial $p(x)$ being of maximum degree $2m + 3$. This means that $|I - I_{SOD}| = 0$ for $I := \int_a^b p(x)dx$. This directly follows from the calculation formula given now, and needs thus not be proven.

Error estimation

Theorem 1 (Error estimation of Simpson Odd Derivative): The quadrature error of the Simpson Odd Derivative Integration is of the order h^{2m+4} :

$$E_{SOD} := |I - I_{SOD}| \leq \frac{\gamma}{2} \cdot h^{2m+4} \cdot (b - a) \cdot \|f^{(2m+4)}\|_{max}. \tag{8}$$

With the error coefficient γ given in Table 1.

Proof: Considering the quadrature error, one has to calculate the Taylor error. The Taylor remainder of order $2m + 4$ of Equation 5 is $(x_i \leq \xi \leq x_i + 2h)$:

$$\frac{2^{2m+5}}{(2m + 5)!} \cdot h^{2m+5} \cdot f^{(2m+4)}(\xi).$$

The remainder of Equation 6 is:

$$\left(\frac{w_2 + 2^{2m+4}w_3}{(2m + 4)!} - \sum_{j=1}^m \alpha_j \cdot \frac{2^{2m-2j+5}}{(2m - 2j + 5)!} \right) \cdot h^{2m+5} f^{(2m+4)}(\xi).$$

The overall error $E_{SOD,i}$ for one interval i from x_i to $x_i + 2h$ is then calculated by the difference of both Taylor remainders:

$$\begin{aligned} E_{SOD,i} &:= |I_i - I_{SOD,i}| \\ &= \left| \left(\frac{2^{2m+5}}{(2m+5)!} - \frac{\omega_2 + 2^{2m+4}\omega_3}{(2m+4)!} + \sum_{j=1}^m \alpha_j \cdot \frac{2^{2m-2j+5}}{(2m-2j+5)!} \right) \cdot h^{2m+5} \cdot f^{(2m+4)}(\xi) \right| \\ &\leq \left| \frac{2^{2m+5}}{(2m + 5)!} - \frac{\omega_2 + 2^{2m+4}\omega_3}{(2m + 4)!} + \sum_{j=1}^m \alpha_j \cdot \frac{2^{2m-2j+5}}{(2m - 2j + 5)!} \right| \cdot h^{2m+5} \cdot |f^{(2m+4)}(\xi)| \\ &=: \gamma \cdot h^{2m+5} \cdot |f^{(2m+4)}(\xi)|. \end{aligned}$$

This gives the calculation for the error coefficient γ . Please note from Table 1, that the factor γ decreases about a factor of 40 for each higher odd derivative used (this factor seems to converge to $4\pi^2 \approx 39, 48$ for higher derivatives, likewise for the Euler- MacLaurin formula, but this is just an assumption) !

For the overall error then holds, using Equation 1:

$$E_{SOD} = \sum_{i=1}^{n/2} E_{SOD,i} \leq \frac{\gamma}{2} \cdot h^{2m+4} \cdot (b - a) \cdot \|f^{(2m+4)}\|_{max}.$$

Boole Quadrature with Odd Derivatives

When one uses not two neighbouring intervals but four, the quadrature formula referred to Boole results. For this, the integral of f can be calculated by:

$$I_B := h \sum_{i=0}^{\frac{n}{4}-1} w_1 f(x_{4i}) + w_2 f(x_{4i} + h) + w_3 f(x_{4i} + 2h) + w_4 f(x_{4i} + 3h) + w_4 f(x_{4i} + 4h), \tag{9}$$

With the weights $w_1 = w_5 = \frac{14}{45}$, $w_2 = w_4 = \frac{64}{45}$ and $w_3 = \frac{24}{45}$. The error of Boole’s rule is converging by the order of h^6 , two orders of magnitude faster than Simpson’s rule:

$$E_B := |I - I_B| \leq \frac{2}{945} \cdot h^6 \cdot (b - a) \cdot \|f^{(6)}\|_{max}.$$

Definition of Boole Odd Derivative

Like for the Simpson rule, Boole’s rule can also be extended by the odd derivatives at the borders, leading to the so-called “Boole Odd Derivative Quadrature BOD”.

Definition 2 (Boole Odd Derivative Quadrature BOD): Given the integral $I := \int_a^b f(x)dx$ and Boole’s Rule I_B defined by Equation 9 with n doubly even and equidistant dissections of the interval of width h (defined by Equation 1) of the function $f \in C^{2m+6}[a, b]$, with a natural number $m \geq 1$. Given also the odd derivatives of f at the borders with $|f^{(2j-1)}(a)| < \infty$ and $|f^{(2j-1)}(b)| < \infty$ for $j = 1, \dots, m$ with at least one $f^{(2j-1)}(a)$ or $f^{(2j-1)}(b) \neq 0$.

A fast converging formula for the numerical value of the integral is then calculated by:

$$I_{BOD} := I_B + \sum_{j=1}^m \alpha_j \cdot h^{2j} \cdot (f^{(2j-1)}(a) - f^{(2j-1)}(b)). \tag{10}$$

The weights w_i differ compared to Equation 9 with the number of derivatives m being used. They are shown together with the α_j in Table 2. The result for $m = 1$ is also given by Burg in [6]. In this paper we give a general approach for any odd derivative used.

Table 2. Coefficients of the Boole Odd Derivative Quadrature up to $m = 2$ (third derivative); due to numerical instabilities, the calculation of the LSE for $m \geq 3$ is not practical).

m	w ₁ =w ₅	w ₂ =w ₄	w ₃	α ₁	α ₂	Error coef. γ
1	$\frac{434}{945}$	$\frac{1024}{945}$	$\frac{864}{945}$	$\frac{4}{63}$		$\frac{16}{99225}$
2	$\frac{7874}{16065}$	$\frac{16384}{16065}$	$\frac{15744}{16065}$	$\frac{4}{51}$	$-\frac{16}{16065}$	$\frac{3}{782792}$

The approach to calculate the w_i and α_j is identical to SOD, but the numbers slightly change. This time, f is approximated by its Taylor series up to order $2m + 5$. When using the same approach as in Equations 5 and 6 with the Taylor series up to order $2m + 5$, the following LSE with $6 + 2m$ rows and $5 + m$ columns results:

$$\begin{pmatrix} 1 & 1^0 & \dots & 4^0 & 0 & \dots & 0 \\ 0 & 1^1 & \dots & 4^1 & 0 & \dots & 0 \\ 0 & 1^2 & \dots & 4^2 & -\frac{2!}{1!} \cdot 4^1 & \dots & 0 \\ 0 & 1^3 & \dots & 4^3 & -\frac{3!}{2!} \cdot 4^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1^{2m} & \dots & 4^{2m} & -\frac{(2m)!}{(2m-1)!} \cdot 4^{2m-1} & \dots & -\frac{(2m)!}{(1)!} \cdot 4^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1^{2m+5} & \dots & 4^{2m+5} & -\frac{(2m+5)!}{(2m+4)!} \cdot 4^{2m+4} & \dots & -\frac{(2m+5)!}{(6)!} \cdot 4^6 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ \vdots \\ w_5 \\ \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} 4^1 \\ 1 \\ \vdots \\ \frac{4^{2m+6}}{2m+6} \end{pmatrix}$$

The solution of the LSE gives the coefficients w_i and α_j of Table 2. The error of BOD is zero for all polynomials of degree smaller or equal to $2m + 5$.

Error estimation

Theorem 2 (Error estimation of Boole Odd Derivative): The quadrature error of the Boole Odd Derivative Integration is of the order h^{2m+6} :

$$E_{BOD} := |I - I_{BOD}| \leq \frac{\gamma}{4} \cdot h^{2m+6} \cdot (b - a) \cdot \|f^{(2m+6)}\|_{max}.$$

With the error coefficient γ given in Table 2.

Proof: The proof is analogous to the proof for SOD. In this case, the Taylor remainder of order $2m + 6$ has to be considered. The remainder for the integral I from x_i to $x_i + 4h$ is ($x_i \leq \xi \leq x_i + 4h$):

$$\frac{4^{2m+7}}{(2m + 7)!} \cdot h^{2m+7} \cdot f^{(2m+6)}(\xi).$$

The remainder of Equation 10 is:

$$\left(\frac{w_2 + 2^{2m+6}w_3 + 3^{2m+6}w_4 + 4^{2m+6}w_5}{(2m + 6)!} - \sum_{j=1}^m \alpha_j \cdot \frac{4^{2m-2j+7}}{(2m - 2j + 7)!} \right) \cdot h^{2m+7} f^{(2m+6)}(\xi).$$

The overall error $E_{BOD,i}$ for one interval i from x_i to $x_i + 4h$ is calculated by the difference of both Taylor remainders:

$$E_{BOD,i} := |I_i - I_{BOD,i}| \leq \gamma \cdot h^{2m+7} \cdot |f^{(2m+6)}(\xi)|.$$

The error coefficient γ then calculates to:

$$\gamma = \left| \frac{4^{2m+7}}{(2m + 7)!} - \frac{w_2 + 2^{2m+6}w_3 + 3^{2m+6}w_4 + 4^{2m+6}w_5}{(2m + 6)!} + \sum_{j=1}^m \alpha_j \cdot \frac{4^{2m-2j+7}}{(2m - 2j + 7)!} \right|.$$

Here, also, the error coefficient decreases by a factor of about 40 with each odd derivative being used.

The overall error then is, using Equation 1:

$$E_{BOD} = \sum_{i=1}^{n/4} E_{BOD,i} \leq \frac{\gamma}{4} \cdot h^{2m+6} \cdot (b - a) \cdot \|f^{(2m+6)}\|_{max}.$$

Applications

Algorithm: In the following we give the algorithm. It has to be noticed that the algorithm is practical, when the derivatives of f at the borders are easy to calculate, which is usually provided in common programming languages like Python, Mathematica or Matlab. Then, the derivatives at the borders have to be calculated only once and the integral is easily calculated to any accuracy needed. The main advantages are its simple calculation and its very fast

convergence. However, when the derivatives of f are not easy to be determined or its values at the borders become excessively large, the algorithm may not be suited.

Algorithm 1 calculate $I := \int_a^b f(x) dx$

Require:

- 1: ➤ $h = \frac{b-a}{n}$
 - $n \geq 2$ even natural number for SOD
 - $n \geq 4$ doubly even natural number for BOD
 - $m \geq 1$ natural number
 - Vector $\vec{w} = (w_1, w_2, w_3 + w_1, w_2, \dots, w_3 + w_1, w_2, w_1)^T$ of length $n + 1$ for SOD, see Table 1
 - Vector $\vec{w} = (w_1, w_2, w_3, w_4, w_5 + w_1, w_2, \dots, w_4, w_5 + w_1, w_2, w_3, w_2, w_1)^T$ of length $n + 1$ for BOD, see Table 2
 - Vector $\vec{f} = (f(a), f(x_1), \dots, f(x_{n-1}), f(b))^T$ of length $n + 1$, function values at x-values x_i
 - Vector $\vec{\alpha} = (\alpha_1 \cdot h^2, \dots, \alpha_m \cdot h^{2m})^T$ of length m , see Table 1 (SOD) or 2 (BOD)
 - Vector $\vec{f}'_D = (f'(a) - f'(b), \dots, f^{(2m-1)}(a) - f^{(2m-1)}(b))^T$ of length m

Ensure: $I = \int_a^b f(x) dx$

- 2: $I = h \cdot \vec{w} \cdot \vec{f} + \vec{\alpha} \cdot \vec{f}'_D$ (scalar products)

Please note that only SOD or BOD needs to be chosen. The algorithm (both SOD and BOD) uses in sum $3n + m^2 + 8m + 5$ operations, out of which are $n + 1 + 2m$ function calls.

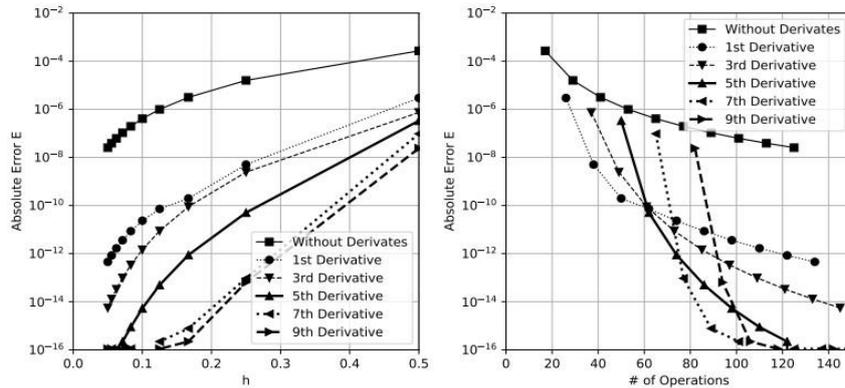
Example 1: Calculation of the Integral of the Gaussian distribution: Now we discuss the approximation of the integral of the Gaussian distribution with $f(x) := \exp(-x^2)$ by Simpson and Boole Odd Derivative Quadrature and compare it with several common quadrature methods. In special, we calculate the integral by using Algorithm 1.

$$I := \int_0^2 e^{-x^2} dx$$

Results for the absolute error of Simpson Odd Derivative Quadrature over step size and number of operations are given in Figure 1. One clearly sees three effects:

- The much faster convergence of SOD compared to the normal Simpson Quadrature without derivatives,
- The much more efficient convergence of SOD, especially when using higher derivatives; machine precision accuracy is given within about 100 operations or only 16 steps ($h = 0.125$),
- The overall very small errors especially for decreasing interval size h .

Figure 1. Absolute error of $I := \int_0^2 e^{-x^2} dx$ over step width h (left image) and the number of operations (right figure) using Simpson Odd Derivative (Equation 4).



When comparing SOD with BOD as given in Figure 2, one sees that Boole Odd Derivative is even more efficient than Simpson Odd Derivative, but the effect is not as strong as expected, showing about a one order of magnitude smaller error with the same number of intervals or operations. However, the effect of the derivatives is evident, as each derivative decreases the error by about a factor of 100, two orders of magnitude.

To further classify the method, we compare it with usual Newton Cotes Quadratures. For this, we calculate $I := \int_0^2 e^{-x^2} dx$ up to an accuracy $<1e-12$. We show how many operations and function calls are necessary to reach this accuracy. As methods we use the Trapezoidal, Simpson and Boole Quadrature without derivatives and SOD and BOD up to $m = 3$. Table 3 shows the results. The efficiency of the use of odd derivatives can clearly be seen. Take e.g. SOD with $m = 3$. According to Table 1 and Equation 4, the error estimation for $m = 3$ (5th derivative) depending on the step width h is:

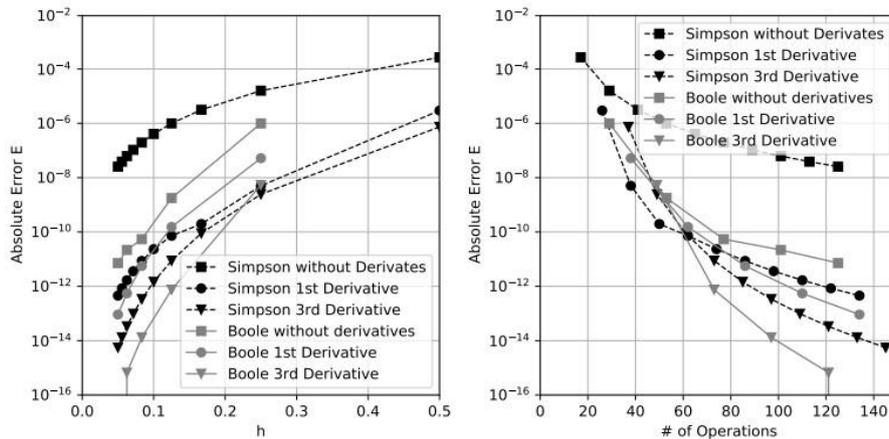
$$E_{SOD,m=3} \leq \frac{h^{10}}{7952175} \cdot \|f^{(10)}\|_{max} \approx 1.26 \cdot 10^{-7} \cdot h^{10} \cdot \|f^{(10)}\|_{max}$$

which leads to a very small error even at a relatively large step width (or small numbers of n). Additionally, very little numbers of operations and function calls are necessary, resulting in a very fast calculation.

Table 3. Approximation of $\int_0^2 e^{-x^2} dx = 0.882081390762422$ to an accuracy $<1e-12$

Rule	Degree m	Intervals n	Abs. error	#op	Function calls
Trapezoidal Rule (Eq. 2)	0	155700	9.84e-13	311400	155700
Simpson's Rule (Eq. 3)	0	506	9.93e-13	1520	507
Boole's Rule (Eq. 9)	0	60	7.54e-13	185	61
SOD (Eq. 4)	1	36	8.43e-13	122	39
SOD (Eq. 4)	2	24	3.34e-13	97	29
SOD (Eq. 4)	3	12	8.83e-13	74	19
BOD (Eq. 10)	1	32	5.50e-13	106	33
BOD (Eq. 10)	2	16	7.50e-13	73	21

Figure 2. Absolute error of $I: = \int_0^2 e^{-x^2} dx$ over step width h (left image) and the number of operations (right figure) using Simpson and Boole Odd Derivative up to $m = 2$.

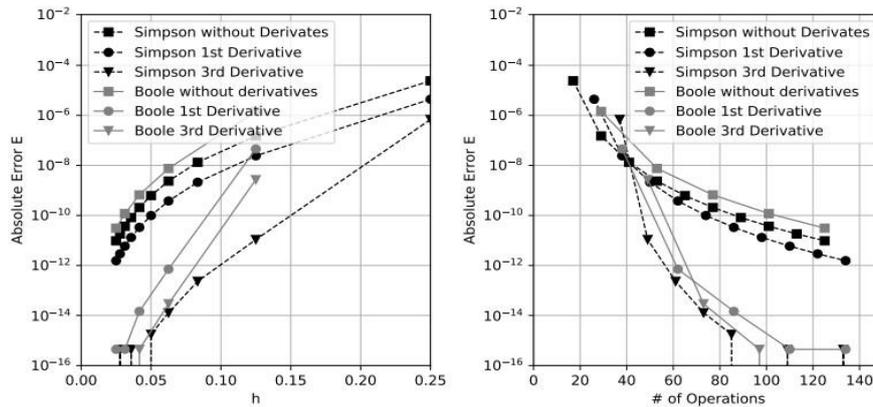


Example 2: Calculation of π : As indicated above, the strength of the method shows especially when both the number of derivatives m and the number of intervals n is increased, which means that the width h is decreased. For the second illustration, we take the function $f(x) = 4/(1 + x^2)$. The primitive of the function is $4 \arctan x$, so for the integral from 0 to 1 follows:

$$\int_0^1 \frac{4}{1 + x^2} dx = \pi$$

This function fulfills all drawbacks concerning the method given above: the higher derivatives are hard to calculate by hand, $f^{(2j-1)}(a) = f^{(2j-1)}(b) = 0$ for $j = 2, 4, 6, \dots$, so the prerequisites are not optimal. Furthermore, the higher derivatives at b increase extremely fast ($f'(b) = -2, f^{(5)}(b) = 60, f^{(9)}(b) = -45360, f^{(13)}(b) = 194594400, \dots$). However; these drawbacks are highly compensated when one uses more intervals. Results are shown in Figure 3, which clearly show the very fast convergence towards π , even when using only the first and third derivative. Surprisingly, Boole Odd Derivative performs even less than Simpson Odd Derivative, probably coming from the ill conditions of the higher derivatives used for BOD.

Figure 3. Absolute error of $I := \int_0^1 \frac{4}{1+x^2} dx = \pi$ over step width h (left image) and the number of operations (right figure) using Simpson and Boole Odd Derivative up to $m = 2$.



CONCLUSION

In this paper, a very efficient and fast converging Quadrature formula for numerical integration has been shown, which is also very easy to implement. Besides of the function value, it also uses the odd derivatives of the function at the borders only. A general method for the calculation of several Newton-Cotes formulas with odd derivatives has been given, which is applicable to all Newton-Cotes Quadratures and shown here only for Simpson’s and Boole’s rule. With each odd derivative additionally used, the convergence increases with the square of the step size h^2 and the absolute error coefficients decrease by about a factor of 40. The error formulas and a method to calculate the errors have also been given, with the error formula being proven.

The algorithm of the method has been given with all its prerequisites. It is very easy to write into program code. The efficiency of the algorithm has been shown using two common examples. Comparing it with usual Newton-Cotes formulas, a much faster convergence is shown, making the method very feasible. However, the method is not suitable in occasions when the derivatives of the function to be integrated are difficult or intricate to calculate. In any other cases, a very practical method is provided.

AVAILABILITY OF SUPPORTING DATA

The code for the algorithm written in Python is available from the author.

ACKNOWLEDGMENTS

The author likes to thank all Python enthusiasts and especially Guido van Rossum for that wonderful programming language.

REFERENCES

1. Davis P, et al. Methods of Numerical Integration. 2007.
2. Schwarz H. Numerische Mathematik (German). 1997.
3. Ujevic N. A generalization of the modified simpson’s rule and error bounds. ANZIAM Journal. 2005;47:1–13.
4. Catinas T, et al. Optimal quadrature formulas based on the ϕ -function method. Studia Univ ”Babes-Bolyai”, MathematicA. 2006;1:13

5. Stroud A, et al. Quadrature formulas with multiple gaussian nodes. J SIAM Numerical Analysis. 1965;B2:129-143.
6. Burg C. Derivative-based closed newton-cotes numerical quadrature. Applied Mathematics and Computation. 2012;218:7052-7065.
7. Burg C, et al. Derivative-based midpoint quadrature Rule. Appl Math. 2013;4:228-234.