ISSN 1022-8594
JUJS

Vol. 42, No. 2, pp.63-78

Comparison of the Methods for Numerical Integration

Md. Siddiqur Rahman*, Mst. Moushumi Pervin and Md. Abul Kalam Azad
Department of Statistics, Jagannath University, Dhaka, Bangladesh

Abstract

Numerical integration methods are generally helpful to determine the integral value of a function for which the integration becomes difficult or when it is impossible to find out the exact integral value. The main objective of this paper is to determine the optimum number of partitions for various numerical integration methods so that these methods can give us the best approximate result. We have used these methods to get the optimum result in such a way that after setting initial partitions, these methods will automatically continue to take more additional partitions until the difference between two successive integral values will be less than the considered tolerance limit. For each of the methods, we have recorded the number of steps to get the optimum number of partitions along with their computational error and average CPU time. Among all the considered methods, Weddle's rule outperformed compared to other methods because it has required less optimum number of steps and less average CPU time, and also produced a small amount of computational error to obtain the approximate integral value. The composite integration method also performed well for a large number of subdivisions. But one limitation of the method is that what number of subdivisions should be taken for a good result is not fixed in advance. After that, the order of the methods which performed well is Boole's rule, Simpson's 3/8 rule, Simpson's rule, Simpson's 1/3 rule, Midpoint rule and Trapezoidal rule respectively.

Keywords: Numerical integration; Optimum number of partitions; Tolerance limit; Computational error; Weddle's rule.

Introduction

Numerical integration is a method of finding a definite integral in absence of closed-form expression for the integral or when difficult to find closed-form integration or for unknown explicit function. More specifically, it consists of methods that help us to find the approximate area under a function plotted on a graph [1]. It has a wide range of applications in engineering, finance, statistics, actuarial science, biostatistics, etc. [1-6] and also used in estimation population abundance as an intermediate tool

*Author for correspondence e-mail: rsiddiq11@stat.jnu.ac.bd

[7]. Numerical integration also known as integral calculus is also needed for reconstruction of medical images [8] or in physics [9].

In [10], three different methods of numerical integration such as midpoint rule, trapezoidal rule, and Simpson's rule were compared by using a few simple functions and calculated values for the various number of subintervals to find out which one is the most accurate and fastest, and the author found Simpson's rule is the best one. The complexity of different types of algorithms was discussed in [11], and also in [12], [13] multidimensional integration ideas were considered. In [6], the authors proposed a new numerical integration method named composite numerical integration and compared it with some Newton-Cotes methods of integration, and found that this proposed method outperformed compared to other methods. The authors in [14] developed a mathematical simulator by combining Trapezoidal rule, Simpson's 1/3 rule and Simpson's 3/8 rule to solve numerical integration problems. In [6] most of the papers while using different numerical integration methods, the researchers used the rules that the number of segments for the Simpson's 1/3 rule must be even, for Simpson's 3/8 rule the number of segments must be a multiple of 3, for Boole's rule the number of segments must be a multiple of 4 and for Weddle's rule the number of segments must be a multiple of 6 and so on. Our objective is to determine the optimum number of segments or partitions for various numerical integration methods such as Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule, Simpson's rule, Boole's rule, Weddle's rule, and Composite integration rule so that it can give us correct results up to a certain decimal place. More specifically, if we use t as the initial number of partitions, then the algorithm will take more additional partitions to obtain the optimum value such that the difference between the two successive values will be less than a certain tolerance limit. Mathematically, suppose the initial number of partitions = t as per any rule in the first step. It will take more s additional steps or $(s \times t)$ partitions to obtain the optimum value. Let the value of the $(s+1)^{th}$ step be w_{s+1} and s^{th} step has the value w_s. After starting with t partitions, the process will stop to take additional partitions when $|w_{s+1} - w_s| < \varepsilon$ where ε is a very small number called tolerance limit. To best of our knowledge, this is the first study for obtaining the optimum number of partitions for various integration methods.

Those functions which can't be integrated analytically are evaluated by using several numerical integration methods such as General Gauss Legendre Quadrature, Newton-Cotes, Romberg integration, and Monte Carlo integration. General Gauss Legendre Quadrature methods use Newton's forward interpolation formula. General Quadrature methods such as the Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule, Boole's rule, and Weddle's rule are special cases of 1st, 2nd, 3rd, 4th and 6th order polynomials respectively.

Given a set of data points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ of a function $y = f(x)$ where $f(x)$ is not known explicitly, it is required to compute the value of the definite integral

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

A general formula for numerical integration can be derived by using Newton's forward interpolation formula. Let the interval $[a, b]$ be divided into *n* equal subintervals such that $a = x_0 < x_1 < \cdots < x_n = b$. So, we can write $x_n = x_0 + nh$.

Approximating $f(x)$ by Newton's forward interpolation formula we can write the above integral as

$$I = \int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_0+nh} ydx = \int_{x_0}^{x_0+nh} \left[\sum_{r=0}^{n} \binom{u}{r} \Delta^r y_0\right] dx \qquad (1)$$

where $x = x_0 + uh$, so $dx = hdu$.When $x = x_0$, $u = 0$ and $x = x_0 + nh, u = n$, then expression (1) can be written as

$$I = h\left[ny_0 + \frac{n^2}{2}\Delta y_0 + \left(\frac{n^3}{3} - \frac{n^2}{2}\right)\frac{\Delta^2 y_0}{2!} + \left(\frac{n^4}{4} - n^3 + n^2\right)\frac{\Delta^3 y_0}{3!} + \cdots\right] \quad (2)$$

The equation (2) is called General Gauss Legendre Quadrature formula. From this general formula, we can obtain different integration formulae by putting $n = 1, 2, 3, ...$, etc.

Existing Methods

A brief description of the existing methods of numerical integration such as Trapezoidal rule, Simpson’s 1/3 rule, Simpson’s 3/8 rule, Boole’s rule, Weddle's rule, Midpoint rule, Simpson's rule, and Composite integration method are given below.

Trapezoidal Rule

Substituting $n = 1$ in the expression (2) and neglecting all differences greater than the first we get

$$I_1 = \int_{x_0}^{x_0+h} f(x)dx = \int_{x_0}^{x_0+h} ydx = h\left(y_0 + \frac{1}{2}\Delta y_0\right) = h\left(y_0 + \frac{y_1 - y_0}{2}\right) = \frac{h}{2}(y_1 + y_0)$$

In general

$$I_r = \int_{x_0+(r-1)h}^{x_0+rh} ydx = \frac{h}{2}(y_{r-1} + y_r),\ r = 1, 2, \dots, n$$

Thus

$$I = \sum_{r=1}^{n} I_r = \frac{h}{2}\sum_{r=1}^{n}(y_{r-1} + y_r) = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + \cdots + y_{n-1})]$$

This is known as Trapezoidal rule and can be applied to any number of subintervals odd or even.

Simpson’s 1/3 Rule

Putting $n = 2$ in the General quadrature formula given by (2) and neglecting all differences greater than second we get

$$I_1 = \int_{x_0}^{x_0+2h} ydx = h\left[2y_0 + 2\Delta y_0 + \left(\frac{8}{3} - 2\right)\frac{\Delta^2 y_0}{2!}\right]$$

$$= h\left[2y_0 + 2(y_1 - y_0) + \frac{1}{3}(y_2 - 2y_1 + y_0)\right]$$

$$= \frac{h}{3}(y_0 + 4y_1 + y_2)$$

In general

$$I_{r/2} = \int_{x_0+(r-2)h}^{x_0+rh} ydx = \frac{h}{3}(y_{r-2} + 4y_{r-1} + y_r), r \epsilon A = (2, 4, 6, \dots, n)$$

where n is even.

Thus

$$I = \sum_{r\epsilon A} I_{\frac{r}{2}} = \frac{h}{3}\sum_{r\epsilon A}^{n}(y_{r-2} + 4y_{r-1} + y_r)$$

$$= \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2})]$$

Simpson's 3/8 Rule

Substituting $n = 3$ in the General Quadrature formula given by (2) and neglecting all differences greater than the third we get

$$I_1 = \int_{x_0}^{x_0+2h} ydx \ = h\left[2y_0 + 2\Delta y_0 + \left(\frac{8}{3} - 2\right)\frac{\Delta^2 y_0}{2!}\right]$$

$$= h\left[2y_0 + 2(y_1 - y_0) + \frac{1}{3}(y_2 - 2y_1 + y_0)\right]$$

$$= \frac{h}{3}(y_0 + 4y_1 + y_2)$$

In general

$$I_{r/3} = \int_{x_0+(r-2)h}^{x_0+rh} ydx = \frac{h}{3}(y_{r-2} + 4y_{r-1} + y_r), r\epsilon A = (3, 6, \dots, n)$$

where *n* is a multiple of 3.

Thus

$$I = \sum_{r\epsilon A} I_{r/3} = \frac{h}{3}\sum_{r\epsilon A}^{n}(y_{r-2} + 4y_{r-1} + y_r)$$

$$= \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2})]$$

Boole's Rule

Substituting $n = 4$ in the General Quadrature formula given by (2) and neglecting all differences greater than the fourth we get

$$I_1 = \int_{x_0}^{x_0+4h} ydx = h\left[4y_0 + 8\Delta y_0 + \frac{20}{3}\Delta^2 y_0 + \frac{2}{3}\Delta^3 y_0 + \frac{14}{45}\Delta^4 y_0\right]$$

$$= \frac{2h}{45}(7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4)$$

In general

$$I_{r/4} = \int_{x_0+(r-4)h}^{x_0+rh} ydx = \frac{2h}{45}(7y_{r-4} + 32y_{r-3} + 12y_{r-2} + 32y_{r-1} + 7y_r)$$

where $r\epsilon A = (4, 8, 12, \dots, n)$ and *n* is multiple of 4.

Thus

$$I = \sum_{r\epsilon A} I_{r/4} = \frac{2h}{45} \sum_{r\epsilon A}^{n} (7y_{r-4} + 32y_{r-3} + 12y_{r-2} + 32y_{r-1} + 7y_r)$$

$$= \frac{2h}{45} [7(y_0 + y_n) + 32(y_1 + y_3 + \cdots + y_{n-1})$$

$$+12(y_2 + y_6 + \cdots + y_{n-2}) + 14(y_4 + y_8 + \cdots + y_{n-4})]$$

Weddle's Rule

Substituting $n = 6$ in the General Quadrature formula given by (2) and neglecting all differences greater than the sixth we get

$$I_1 = \int_{x_0}^{x_0+6h} y dx$$

$$= h\left[6y_0 + 18\Delta y_0 + 27\Delta^2 y_0 + 24\Delta^3 y_0 + \frac{123}{10}\Delta^4 y_0 + \frac{33}{10}\Delta^5 y_0 + \frac{41}{140}\Delta^6 y_0\right]$$

$$= \frac{3h}{10}(y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6)$$

In general

$$I_{r/6} = \int_{x_0+(r-6)h}^{x_0+rh} y\, dx = \frac{3h}{10}(y_{r-6} + 5y_{r-5} + y_{r-4} + 6y_{r-3} + y_{r-2} + 5y_{r-1} + y_r)$$

where $r\epsilon A = (6, 12, 18, \dots, n)$ and *n* is a multiple of 6.

Thus

$$I = \sum_{r\epsilon A} I_{r/6} = \frac{3h}{10} \sum_{r\epsilon A}^{n} (y_{r-6} + 5y_{r-5} + y_{r-4} + 6y_{r-3} + y_{r-2} + 5y_{r-1} + y_r)$$

$$= \frac{3h}{10}[(y_0 + y_n) + 5(y_1 + y_5 + \cdots + y_{n-1}) + (y_2 + y_4 + \cdots + y_{n-2})$$

$$+6(y_3 + y_9 + \cdots + y_{n-3})]$$

Midpoint Rule

Let us assume that $y = f(x)$ is the function which we want to integrate over the interval $[x_0,\ x_0 + nh]$, where *h* is the length of subintervals.

For the Midpoint rule [10], $\left(\frac{y_i + y_{i+1}}{2}\right)$ is used as an approximation for the value in the subinterval. Now we get

$$I = \int_{x_0}^{x_0+nh} ydx$$

$$= h\left[\left(\frac{y_0+y_1}{2}\right) + \left(\frac{y_1+y_2}{2}\right) + \cdots + \left(\frac{y_{n-1}+y_n}{2}\right)\right] = h\sum_{i=0}^{n-1}\left(\frac{y_i+y_{i+1}}{2}\right)$$

Simpson's Rule

Simpson's rule is based on polynomial interpolation and uses second - degree polynomial [10]. In this rule, y_i, y_{i-1} and h are used as three sides of the figure, but the fourth side is parabola approximated to the graph of the function. Values at the beginning and end of the subinterval are used as the points needed to approximate this parabola. Now, the value of the area for $[x_{i-1}, x_i]$ is calculated by the following formula

$$I_i = f(x_i) + 4f\left(x_i + \frac{h}{2}\right) + f(x_i + h) = y_i + 4y_{i+\frac{h}{2}} + y_{i+h}$$

where $i = 1, 2, \ldots, n$ and $h = \frac{x_n - x_0}{n}$. The whole integral gives the following approximated value

$$I = \int_{x_0}^{x_0+nh} ydx = \frac{1}{6}\sum_{i=1}^{n} hI_i = \frac{h}{6}\sum_{i=1}^{n}\left[y_i + 4y_{i+\frac{h}{2}} + y_{i+h}\right]$$

Composite Numerical Integration Method

The composite numerical integration method [6] estimates the area under the curve of a function *f* numerically and above the horizontal axis between the interval $[\,a, b]$. It can be obtained by summing the areas of all the *n* partitions each of width $h = \frac{b-a}{n}$ and k sub-divisions. Suppose $y_i = f(x_i), i = 0, 1, 2, \ldots, n$ be the ordinate at x_i of the function *f*. Also assume that the interval $[x_i, x_{i+1}]$ is divided into k equidistant points $x_i + \frac{t}{k}h, t = 1, 2, \ldots, k$; then the corresponding ordinates of f are given by $y_{i+\frac{t}{k}} = f\left(x_i + \frac{t}{k}h\right)$, $t = 1, 2, \ldots, k; i = 0, 1, 2, \ldots, n-1$. Thus the composite numerical integration method proposed by Felix et al. (2016) is given by

$$I_n = \frac{h}{2k}\left[y_0 + 2\sum_{i=1}^{n-1} y_i + 2\sum_{i=0}^{n-1}\sum_{t=1}^{k-1} y_{i+\frac{t}{k}} + y_n\right]$$

where the subscript in I_n means that the area estimation is based on n segments in the interval $[a, b]$.

Results and Discussions

Numerical integration is the process of finding the area under a function especially when the integration of the function becomes very difficult or impossible to find the integrals. Due to this reason, we use different types of integration methods as a substitute for the exact integration method. As we want to know which methods give us the approximate result close to the exact integration result, we will use several simple functions so that these functions can be integrated easily, and then we will compare the results of the exact integration method with different types of integration methods to determine the most appropriate one for estimating the integrals. Initially, we will set up a tolerance limit 10^{-9} for all the integration methods. We will record the CPU time taken by each method to get the desired integral value and repeat this process 100 times. Average CPU time will be obtained by averaging these CPU times. The errors of integral values obtained by different integration methods are calculated by the following formula

$$Error\ (in\ percentage) = \frac{Exact - Approximate}{Exact} \times 100$$

Example 1: As an illustration, we will use the following integral problem

$$\int_0^2 e^{x^2} dx = \int_0^2 \left(\sum_{n=0}^{\infty} \frac{x^{2n}}{n!}\right) dx$$

$$= \sum_{n=0}^{\infty} \left(\int_0^2 \frac{x^{2n}}{n!} dx\right)$$

$$= \sum_{n=0}^{\infty} \frac{2^{2n+1}}{(2n+1)n!}$$

$$= 2 + \frac{2^3}{3} + \frac{2^5}{5(2!)} + \cdots + \frac{2^{201}}{201(100!)} + \cdots$$

$= 16.45262777$ (As the value is taken up to 8 decimal places)

The main objective of this paper is to determine the optimum number of partitions n at which the integration methods give us the approximate result. So, initially we have chosen the primary value of n for different integration methods. As we can see from Table 1 that initially we have taken $n = 1$ for Trapezoidal rule. After successful completion of 5262 steps (5262 partitions), this method gives the approximate result 16.4526304 with error 1.59861825×10^{-5}(in percentage) and it takes an average of 3.5414 CPU time to complete the iteration task. This method has taken 5261 additional steps (5261 additional partitions) automatically and stopped the procedure of taking additional partitions because the difference between the integral values after 5262 partitions and 5261 partitions is less than the tolerance limit 10^{-9}. If we increase the value of the tolerance limit to minimize the error, we will need more steps as well as more partitions to get the desired result and vice versa. For example, if the tolerance limit is set up as 10^{-12}, then it will need 52342 steps (52342 partitions) to get 16.45262779 as the integral value with error 1.6150918×10^{-7}. Here we can see that it will reduce the error but it will take more partitions as well as more CPU time. We initially set up $n = 2$ for Simpson's 1/3 rule. After 162 steps ($162 \times 2 = 324$ partitions), this method gives the approximate result 16.45262781 with error 2.414301817×10^{-7} so that the difference between the integral values after 322 partitions and 324 partitions is less than the tolerance limit 10^{-9}. Among the methods of integration, we can see that Weddle's rule requires 21 steps (126 partitions) along with error 2.146778994×10^{-8} which are less than any other methods. Also, it takes less CPU time to complete the iteration task compared to other methods. For the composite integration method initially, we set up $n = 500$ partitions and $k = 5000$ sub-divisions with tolerance limit 10^{-9}, then it takes 502 steps to get the approximate result 16.45262777 with less error than any other methods. But if we take a small number of partitions $n = 50$ in the initial step with $k = 500$ sub-divisions, it takes 86 steps to get the approximate value 16.45262777 with error 2.508318815×10^{-7}. By comparing all three results of composite integration, it can be noticed that the error percentages increased as the number of subdivisions k decreased and the percentages of error don't depend on the initial number of partitions n.

Table 1: Estimates of the area and number of steps for example 1 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	16.4526304	1.59861825×10^{-5}	5262	5262 ×1 = 5262	3.5414
Simpson's 1/3 rule	16.45262781	2.414301817×10^{-7}	162	162 × 2 = 324	0.0241
Simpson's 3/8 rule	16.4526278	2.046526199×10^{-7}	138	138 × 3 =414	0.01
Boole's rule	16.45262777	3.130561554×10^{-8}	31	31 × 8 = 248	0.0048
Weddle's rule	16.45262777	2.146778994×10^{-8}	21	21 × 6 = 126	0.0045
Midpoint rule	16.45262568	1.26922428×10^{-5}	4176	4176×1 = 4176	0.01
Simpson's rule	16.45262781	2.414301385×10^{-7}	162	162×1 = 162	0.0291
Composite Integration method	16.45262777 when $n = 500$, $k = 5000$	7.08053955×10^{-11}	502	502	0.5138
	16.45262788 when $n = 500$, $k = 50$	7.079480164×10^{-7}	502	502	0.0417
	16.45262781 when $n = 50$, $k = 500$	2.508318815×10^{-7}	86	86	0.18

For the computational purpose, if we initially set $n = 6^9$ (large n matches with all rules), then we can see from Table 2 that all the methods give us the approximate result after 2 steps only along with very small error except composite integration method. The Composite Integration takes 10077698 steps to get 16.45262777 with error $9.501181458\times10^{-13}$ and also requires an infinite amount of CPU time to finish the task. Also, all other methods require unusual CPU time to complete the task. That's why, it will not be wise to set up a very large value of n at the initial step.

Table 2: Estimates of the area and number of steps for example 1 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	16.45262777	$4.318718845\times10^{-12}$	2	2×1 = 2	3.2019
Simpson's 1/3 rule	16.45262777	$4.318718845\times10^{-14}$	2	2×2 = 4	5.1442
Simpson's 3/8 rule	16.45262777	$4.318718845\times10^{-14}$	2	2×3 = 6	6.2312
Boole's rule	16.45262777	$4.318718845\times10^{-14}$	2	2 × 8 = 16	8.6124
Weddle's rule	16.45262777	$4.318718845\times10^{-14}$	2	2 × 6 = 12	6.6452
Midpoint rule	16.45262777	$5.463179339\times10^{-12}$	2	2 × 1 = 2	0.01
Simpson's rule	16.45262777	$3.217445539\times10^{-12}$	2	2 × 1 = 2	47.1293
Composite Integration method	16.45262777 when $n = 6^9$, $k = 50$	$9.501181458\times10^{-13}$	10077698	10077698	∞

Example 2: For the comparison purpose, we will consider the following integral problem

$$\int_1^2 x\ln(x)\,dx = \left[\frac{x^2\ln(x)}{2} - \frac{x^2}{4}\right] = 0.6362943611$$ (as the value is taken up to 8 decimal places)

From Table 3 it can be seen that after completion of 488 partitions, Trapezoidal rule gives its best approximate result 0.6362946047 with error 3.827614448×10^{-5}. This error is comparatively greater than rest of the considered methods as the tolerance limit was 10^{-9}. The CPU time is 0.0361. Among all the other considered methods, we can see that the Midpoint rule produces the second-highest error 3.030638411×10^{-5} by giving the approximate value of 0.6362941683 as the difference between the integral values after 32 partitions and 34 partitions is less than the tolerance limit 10^{-9}. Also, we can see from Table 3 that Weddle's rule gives us the best approximate result 0.6362943613 after the successful completion of 24 partitions with fewer errors 3.0263381×10^{-8} compared to other methods except for the composite integration method. In the composite integration method, it can be seen that when we set up the initial partitions $n = 50$ and subdivisions $k = 5000$, then this method takes 502 steps and produces less error $1.465653945\times10^{-12}$ which is less than all other methods but it will require more CPU time 0.481 second.

Table 3: Estimates of the area and number of steps for example 2 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	0.6362946047	3.827614448×10^{-5}	488	488×1 = 488	0.0361
Simpson's 1/3 rule	0.6362943651	6.239542542×10^{-7}	17	17 × 2 = 34	0.0016
Simpson's 3/8 rule	0.6362943641	4.73017669×10^{-7}	15	15 × 3 = 45	0.0019
Boole's rule	0.6362943618	1.070230107×10^{-7}	4	4 × 8 = 32	0.0005
Weddle's rule	0.6362943613	3.0263381×10^{-8}	4	4 × 6 = 24	0.0008
Midpoint rule	0.6362941683	3.030638411×10^{-5}	388	388 × 1 = 388	0.01
Simpson's rule	0.6362943651	6.239542542×10^{-7}	17	17 × 1 = 17	0.0016
Composite Integration method	0.6362943611 when $n = 500$, $k = 5000$	$1.465653945\times10^{-12}$	502	502 × 1 = 502	0.481
	0.6362943612 when $n = 50$, $k = 500$	1.452471784×10^{-8}	52	52 × 1 = 52	0.0088

Table 4 indicates that when $n = 6^9$ partitions are set up initially all methods give us the approximate result by producing less computational error but it takes unusual CPU time on average as in example 1 which is presented in Table 2. Simpson's rule takes the most unusual average CPU time 41.4759 second among all other methods. The result of the composite integration method is not included in the following tables as it shows poor performance for a large value of n.

Table 4: Estimates of the area and number of steps for example 2 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	0.6362943611	$8.724130626\times10^{-14}$	2	2 × 1 = 2	2.46
Simpson's 1/3 rule	0.6362943611	0	2	2 × 2 = 4	3.4315
Simpson's 3/8	0.6362943611	0	2	2 × 3 = 6	4.1814

rule					
Boole's rule	0.6362943611	0	2	2 × 8 = 16	4.3619
Weddle's rule	0.6362943611	0	2	2 × 6 = 12	4.619
Midpoint rule	0.6362943611	$1.090516328\times10^{-11}$	2	2 × 1 = 2	0.01
Simpson's rule	0.6362943611	$1.095750807\times10^{-11}$	2	2 × 1 = 2	41.4759

Example 3: The following integral problem will also be considered for our illustration purpose

$$\int_1^5 \frac{1}{x} dx = [\log x] = 1.609437912$$

As it is observed from Table 5 that Trapezoidal rule takes the largest number of partitions 1369 and produces more computational error compared to other methods for obtaining the approximate result. The rest of the interpretations of Table 5 are similar as in Table 1 and Table 3.

Table 5: Estimates of the area and number of steps for example 3 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	1.609438596	4.249747827×10^{-5}	1369	1369×1 = 1369	0.054
Simpson's 1/3 rule	1.609437931	1.162952897×10^{-6}	74	74 × 2 = 148	0.0044
Simpson's 3/8 rule	1.609437928	9.929445612×10^{-7}	63	63 × 3 = 189	0.0065
Boole's rule	1.609437916	2.339011664×10^{-7}	20	20 × 8 = 160	0.0024
Weddle's rule	1.609437915	1.554849827×10^{-7}	14	14 × 6 = 84	0.0012
Midpoint rule	1.60943737	3.371669721×10^{-5}	1087	1087×1 = 1087	0.01
Simpson's rule	1.609437931	1.162952925×10^{-6}	74	74 × 1 = 74	0.0086
Composite Integration method	1.609437912 when $n = 500$, $k = 5000$	$1.258232319\times10^{-11}$	502	502 × 1 = 502	0.1089
	1.609437914 when $n = 50$, $k = 500$	1.272493389×10^{-7}	52	52 × 1 = 52	0.0038

Detail descriptions of Table 6 are similar as in Table 2 and Table 4.

Table 6: Estimates of the area and number of steps for example 3 using different integration methods

Integration method	Obtained value	Error in percentage	No. of steps	No. of partitions	Elapse computing time (second)
Trapezoidal rule	1.609437912	$7.174131644\times10^{-13}$	2	$2 \times 1 = 2$	1.2222
Simpson's 1/3 rule	1.609437912	$6.898203504\times10^{-14}$	2	$2 \times 2 = 4$	1.4102
Simpson's 3/8 rule	1.609437912	$8.277844204\times10^{-14}$	2	$2 \times 3 = 6$	1.2423
Boole's rule	1.609437912	$5.518562803\times10^{-14}$	2	$2 \times 8 = 16$	1.341
Weddle's rule	1.609437912	$6.898203504\times10^{-14}$	2	$2 \times 6 = 12$	1.554
Midpoint rule	1.609437912	1.96874728×10^{-14}	2	$2 \times 1 = 2$	0.01
Simpson's rule	1.609437912	1.9287377×10^{-14}	2	$2 \times 1 = 2$	56.1543

Conclusions

In this paper, we have used several numerical integration methods in such a way that one can get the approximate integral values through an extensive test for various numerical functions. We have considered eight different numerical integration methods such as Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule, Boole's rule, Weddle's rule, Midpoint rule, Simpson's rule and Composite integration method for our comparison purpose. We have set up initial partitions and a tolerance limit for all the considered methods. After that these methods automatically have continued to take additional partitions for obtaining the best approximate integral value until the difference between two successive values of each method is less than the tolerance limit. For comparison purposes, we have also set up initial partitions $n = 6^9$ and noticed that these methods give the approximate result only in two steps but it takes unusual CPU time on average to complete the iteration procedure. That's why initially it will not be wise to set up large partition values. It has been noticed that Weddle's rule gives us comparatively better results compared to other methods because it takes a small number of optimum steps and partitions to get the approximate result, and also produces a less computational error and takes less average CPU time to complete the task. The second method which performed comparatively well is Boole's rule as it takes a fewer number of steps as well as produces a less computational error. Composite integration method has also produced less computational error but it only works for a large number of subdivisions, and also what should be the exact number of subdivisions is not fixed beforehand. So, the composite integration method performed well except that limitation. After that, the chronologies of the methods which performed well are Simpson's 3/8 rule, Simpson's rule, Simpson's 1/3 rule, Midpoint rule and Trapezoidal rule.

References

[1] A. Kaw and M. Keteltas, Lagrange interpolation (2009). http://numericalmethods.eng.usf.edu.

[2] R. L. Burden and J.D Fairs, Numerical analysis. 7th ed. Brooks/Cole Thomson, Pacific Grove, CA ,2001.

[3] R. P. Canale and S. C. Chapra, Numerical methods for engineers: With software and programming applications. McGraw-Hill, New York. 2002.

[4] *D. C. M. Dickson*, *M. R. Hardy* and *H. R. Waters*, Actuarial mathematics for life contingent risks. 2nd ed. Cambridge University Press, New York. 2013.

[5] S.A. Klugman, H.H. Panjer and G.E. Willmot, Loss models: From data to decisions. 2nd ed. John Wiley &Sons, Inc. Publication ,2004.

[6] F.O. Mettle, E. N. B. Quaye, L. Asiedu and K. A. Darkwah, A Proposed Method for Numerical Integration. *British Journal of Mathematics & Computer Science*, 2016, 17(1), 1-15.

[7] G. C. White, E. G. Cooch, Population abundance estimation with heterogeneous encounter probabilities using numerical integration. *The Journal of Wildlife Management*, 2017, 81(2), 322–336.

[8] W. Wei, B. Zhou, D. Połap and M. Wozniak, A regional adaptive variational pde model for computed tomography image reconstruction. *Pattern Recognition*, 2019, vol. 92, pp. 64–81.

[9] M. Liang and T. E. Simos, A new four stages symmetric two-step method with vanished phase-lag and its first derivative for the numerical integration of the schrödinger equation. *Journal of Mathematical Chemistry,* 2016, 54(5), 1187–1211.

[10] A. Winnicka. Comparison of Numerical Integration Methods. Institute of Mathematics. Silesian University of Technology, Poland.

[11] E. Novak, Some results on the complexity of numerical integration, in Monte Carlo and Quasi-Monte Carlo Methods. *Springer*, 2016, 161–183.

[12] A. P. Nagy and D. J. Benson, On the numerical integration of trimmed isogeometric elements. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284, 165–185.

[13] V. Keshavarzzadeh, R. M. Kirby and A. Narayan, Numerical integration in multiple dimensions with designed quadrature. *SIAM Journal on Scientific Computing*, 2018, 40(4), A2033–A2061.

[14] B.K. Dutta, N. Rahman, R.C. Bhowmik, S.Paul, M.R. Kabir and U. Roy, A Numerical Simulator for Solving Numerical Integration. *International Journal of Scientific Engineering and Technology*, 2014, 3(4), 342-345.