Research Article *http://dx.doi.org/10.4314/mejs.v11i2.8*

Developing and Analyzing Newton – C'otes Quadrature Formulae for Approximating Definite Integrals- AC++ Approach

Tuemay Teka*and Dessalegn Assefa
Department of Mathematics, CNCS, Mekelle University, P.O. Box 231, Mekelle, Ethiopia (*tuemay.teka@mu.edu.et).

ABSTRACT

In this paper, different Newton – C'otes quadrature formulae for the approximation of definite integrals and their error analysis are derived. The order of convergences of the methods is also derived and of these Newton – C'otes quadrature formulae, the Simpson's 1/3 rule have been shown to have high order of convergence. Since the functionality of these numerical integration methods is practical only if we can use computer programs and applications to produce approximate solutions with acceptable errors within short period, C++ programs for the selected methods are written. These programs are used on the comparison of the Newton – C'otes quadrature formulae and the result obtained based on the inputs and outputs of the programs for different integrands. The results of these programs show that the convergence of the methods highly depends on the number of iterations. The results of different numerical examples show that for high accuracy of the trapezoidal rule computational effort is higher and round off errors with large number of iterations limit the accuracy. The results show that the Simpson's 1/3 rule produces much more accurate solution than other methods even within small number of iterations. This shows that the error for Simpson's rule 1/3 converges to zero faster than the error for the trapezoidal rule as the step size decreases. It is finally observed that Simpson's 1/3 rule is much faster than the Trapezoidal and the Simpson's 3/8 rules according to the results of the C++ programs.

Keywords: Newton- C'otes quadrature formulae, Interpolation polynomials, Convergence, Error analysis, Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule, C++ programming.

1. INTRODUCTION

Quadrature formulae are algorithms for approximating definite integrals by means of the values of the integrand at a finite number of points. Often need arises for evaluating the definite integral of functions that does not have explicit anti-derivative, in other circumstances the function is not known explicitly but is given empirically by a set of measured or tabulated values. These Quadrature formulae which are based on uniformly spaced points (abscissas) are called Newton – C'otes quadrature formulae (Rao, 2010; Kreyszig, 2006). In certain circumstances where the integral cannot be evaluated analytically, numerical integration is used to give approximate solution to the definite integral. Since solutions obtained in numerical integration are

Submitted on: 29-12-2018Accepted on: 04-04-2019

approximate, there is usually associated error of approximation which is a measure of the deviation of the approximate solution from the exact value (Chapra, 2012).

The most common quadrature formulae have the form

$$\mathrm{I} = \int_a^b w(x)f(x)dx, \qquad (1.1)$$

Where, the weight functionw(x)>0in a closed interval [a,b] and we assume that $w(x)f(x)$ are integrable, in the Riemann sense in [a,b] and the limits of integration is finite (Levy, 2010; Jain et al., 2004).

The need for developing numerical integration methods is that there are definite integrals in which the integrands are known explicitly but cannot be evaluated by applying the analytical techniques in other cases the integrands of the definite integrals are not explicitly known but merely specified by a table of numerical values at some points only and hence analytical methods cannot be applied (Rao, 2010; Burden and Douglas, 2010).

Since, all numerical methods only give approximate solutions, there is a need for their error analysis so as to attain better accuracy and most of these methods cannot be used manually for practical applications due to these associated errors unless we use computer applications (Jain et al., 2004; Chapra, 2012).

Numerical integration is the study of how the numerical value of an integral can be found. Also called quadrature, which refers to finding a square whose area is the same as the area under a curve, it is one of the classical topics of numerical analysis. Of central interest is the process of approximating a definite integral from values of the integrand when exact mathematical integration is not available. Many methods are available for approximating the integral to the desired precision in Numerical integration (Jain et al., 2004; Levy, 2010; Amos and Subramaniam, 2013).

Numerical integration is the process of computing the value of a definite integral from a set of numerical values of the integrand. The process of evaluation of integration of a function of a single variable is sometimes called Mechanical Quadrature. The computation of a double integral of a function of two independent variables is called Mechanical Cubature (Rao, 2010; Sastry, 2012).

The Trapezoidal rule is one of a family of formulas for numerical integration called Newton–Cotes formulae, of which the midpoint rule is similar to the trapezoid rule. Simpson's rule is another member of the same family, and in general has faster convergence than the trapezoidal rule for functions which are twice continuously differentiable, though not in all specific cases (Alomari and Dragomir, 2014; Weidman, 2002). However, for various classes of rougher functions (ones with weaker smoothness conditions), the trapezoidal rule has faster convergence in general than Simpson's rule. Moreover, the trapezoidal rule tends to become extremely accurate when periodic functions are integrated over their periods, which can be analyzed in various ways (Weidman, 2002; Mathews and Kurtis, 2004).

It follows that if the integrand is concave up (and thus has a positive second derivative), then the error is negative and the trapezoidal rule overestimates the true value. This can also be seen from the geometric picture: the trapezoids include all of the area under the curve and extend over it. Similarly, a concave-down function yields an underestimate because area is unaccounted for under the curve, but none is counted above (Burg, 2012; Weidman, 2002).

2. METHODS AND MATERIALS

In this paper, the Newton–C'otes quadrature methods based on Newton's interpolation polynomials are derived and analyzed.

In developing and analyzing the Newton – C'otes quadrature formulas, the following steps will be adopted.

- Taking some samples of tabulated functional values of the integrand,
- Deriving interpolation polynomial which agrees at the tabulated values of the integrand,
- Replacing the integrand by the interpolation polynomial,
- Integrate this polynomial on the interval [a,b],
- Divide the interval into smaller sub intervals and take the sum as the approximate value of the integral,
- Write C++ programs for each of the methods and implement it.

The method of numerical integration which uses interpolation polynomials to approximate the given integral is called Newton – C'otes quadrature formula. The integration formulas of Newton and cotes formulas are obtained if the integrand is replaced by a suitable interpolating polynomial $P(x)$ and if $\int_a^b P(x)dx$ is taken as an approximate value for $\int_a^b f(x)dx$. Consider a uniform partition of the closed interval $[a\,,b]$, given by $x_i = a + ih$, $i = 0, 1, 2, \cdots, n$, where h is the step length given by $h = \frac{b-a}{n}$ and let P_n be an interpolating polynomial of degree n or less satisfying the following conditions, $P_n(x_i) = f(x_i)$, for all $i = 0, 1, 2, \cdots, n$,ad $x_i \neq x_j$ for $i \neq j$.Now using the Newton's interpolation formulae we replace the integrand of the integral and we find the following quadrature formulae.

2.1. Quadrature Methods Based on Newton's Backward Difference Interpolations

In this method we replace the integrand above by Newton's backward difference interpolating polynomial and integrate it with in the limits of integration. The n-th order Newton's backward ifference interpolation formula is given by Rao (2010); and Kharab & Guenther (2001).

$$P_n(x) = f_n + p\nabla f_n + \frac{1}{2}p(p+1)\nabla^2 f_n + \cdots + \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!}\nabla^n f_n \qquad (2.1)$$

Where, $p = \frac{x-x_n}{h}$,$h\ is\ step\ size$ and $f_i = f(x_i) = y_i$, $i = 0,1,2, \dots, n$.

Substituting the formula equation (2.1) for the integrand in the formula equation (1.1) and simplifying we obtain,

$$\int_a^b w(x)f(x)dx = \int_a^b P_n(x)dx \ = h\int_{-n}^0 P_n(p)dp$$

$$= -nh\left[-f_n + \frac{n}{2}\nabla f_n + \frac{n}{12}(3-2n)\nabla^2 f_n + \frac{n}{24}(n-2)(n-2)\nabla^n f_n + \cdots\right] \qquad (2.2)$$

Where, and $\nabla f_n = f_n - f_{n-1}$, and $\nabla^n f_k = \nabla^{n-1} f_k - \nabla^{n-1} f_{k-1}$

This expression gives us different Newton-C'otes quadrature formulas on substituting different values of n.

2.1.1. Trapezoidal Rule

Here, we let $n = 1$in the above equation (2.2) and obtained the following result

$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)]$, Where, $f(x_i) = f_i = y_i$ for all$i = 0, 1, 2, \cdots, n$ .Graphically.

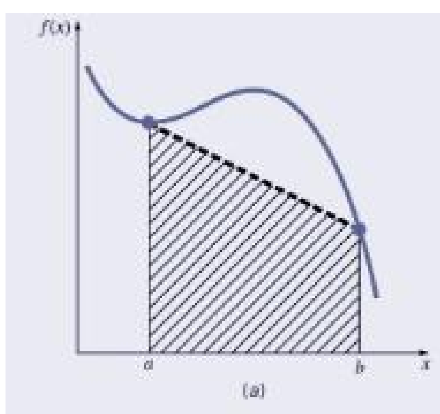

Figure 1. The Trapezoidal rule (Levy, 2010).

Now, the error of this method can be approximated in the following way.

Let the function $y = f(x)$ be continuous and possess a continuous derivatives in$[x_0, x_n]$.

Expanding y in Taylor's series about$x = x_0$, we obtain

$$\int_{x_0}^{x_1} f(x)dx = \int_{x_0}^{x_1}[f_0 + (x - x_0)f_0' + \frac{1}{2}(x - x_0)^2 f_0" + \cdots]dx \tag{2.3}$$

$$= hf_0 + \frac{h^2}{2}f_0' + \frac{1}{6}h^3 f_0" + \ \ldots \quad \text{, Where, } h = (x - x_0)\ .$$

Similarly,

$$\frac{h}{2}[f_0 + f_1] = hf_0 + \frac{h^2}{2}f_0' + \frac{1}{4}h^3 f_0" + \ \ldots. \tag{2.4}$$

Then from (2.3) and (2.4) we obtain

$$E = \int_{x_0}^{x_1} ydx - \frac{h}{2}[f_0 + f_1] = -\frac{1}{12}h^3 f_0" + \cdots.$$

Hence, the error on this interval is given by$E = -\frac{1}{12}h^3 f_0" + \cdots$.

Similarly, we drive the composite trapezoidal rule as follows.

Given an interval $[a, b]$ and a partition $\{x_0, \cdots, x_n\}$ of $[a, b]$ where, $x_0 = a, x_n = b$.

We want to evaluate the definite integral over the whole partition. But, in the previous section we obtain that the approximation of the integral $\int_{x_0}^{x_1} ydx = \frac{h}{2}[y_0 + y_1]$ and similarly we obtain $\int_{x_1}^{x_2} ydx = \frac{h}{2}[y_1 + y_2]$, $\cdots$, and $\int_{x_{n-1}}^{x_n} ydx = \frac{h}{2}[y_{n-1} + y_n]$. Hence the integration over all subdivisions is determined to be

$$\int_{x_0}^{x_n} ydx = \frac{h}{2}[y_1 + 2(y_2 + \cdots + y_{n-1}) + y_n].$$

Graphically,

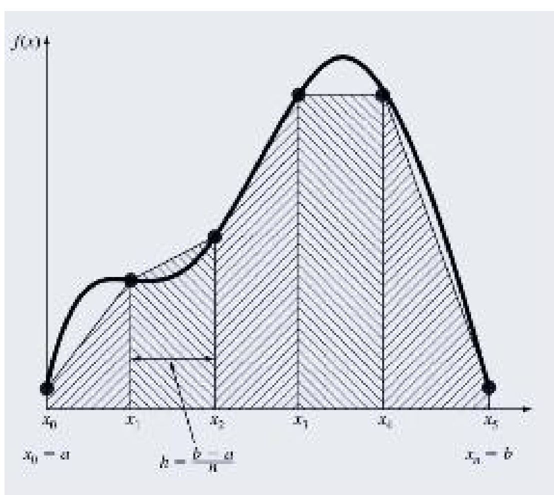

Figure 2. The Composite Trapezoidal rule (Levy, 2010).

Proceeding in a similar manner we obtain the errors in the subintervals $[x_0, x_1], [x_1, x_2], [x_2, x_3], \cdots, [x_{n-1}, x_n\,]$ and adding all the errors we get

$$E_T = -\frac{1}{12}h^3[f_0" + f_1" + \cdots f_{n-1}"]\ .$$

Now assuming that $f"(\xi)$ is the largest value of the n-quantities on the right hand side we obtain

$$E_T = -\frac{1}{12}h^3 n f"(\xi).$$

$$= -\frac{1}{12}(b-a)h^2 f"(\xi)\ \text{, this is true since, } h = \frac{b-a}{n}\ .$$

2.1.2. Simpson's 1/3-Rule

Here, we let $n = 2$ in the above equation (2) and obtain the following result

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)]\ ,$$

Where, $f(x_i) = f_i = y_i$ for all$= 0, 1, 2, \cdots, n$.

Graphically,

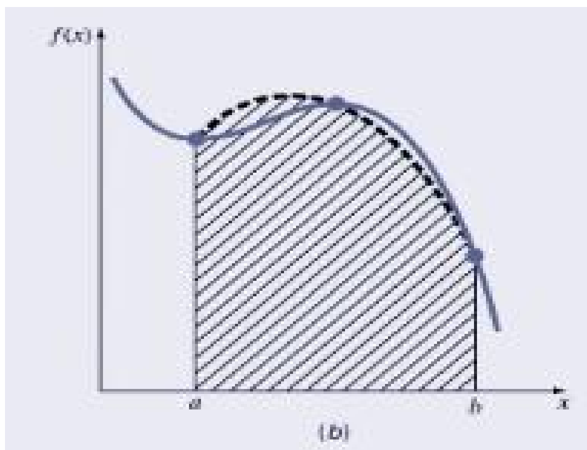

Figure 3. The Simpson's 1/3 rule (Levy, 2010).

Now, the error of this method can be approximated in the following way.

Let the function $y = f(x)$ be continuous and possess a continuous derivatives in $[x_0, x_n]$

Expanding y in Taylor's series about $x = x_0$, we obtain

$$y(x) = y_0 + (x - x_0)y_0' + \frac{1}{2}(x - x_0)^2 y_0'' + \frac{1}{3!}(x - x_0)^3 y_0''' + \cdots$$

$$\int_{x_0}^{x_2} f(x)dx = \int_0^2 h\left[y_0 + phy_0' + \frac{1}{2}(ph)^2 y_0'' + \frac{1}{3!}(ph)^3 y_0''' + \frac{1}{4!}(ph)^4 y_0^{(iv)} + \cdots\right]dp,$$

Where, $x = x_0 + ph$.

$$= h\left[py_0 + \frac{p^2}{2}hy_0' + \frac{1}{6}(ph)^2py_0'' + \frac{1}{24}(ph)^3py_0''' + \frac{1}{120}(ph)^4py_0^{(iv)} + \cdots\right]_0^2.$$

$$= 2hy_0 + 2h^2y_0' + \frac{4}{3}h^3y_0'' + \frac{2}{3}h^4y_0''' + \frac{4}{15}h^5y_0^{(iv)} + \cdots . \quad (2.5)$$

Therefore,

$$y_0 = y_0 \quad (2.6)$$

$$y_1 = y_0 + (x_1 - x_0)y_0' + \frac{1}{2}(x_1 - x_0)^2y_0'' + \frac{1}{3!}(x_1 - x_0)^3y_0''' + \frac{1}{4!}(x_1 - x_0)^4y_0^{(iv)} + \cdots$$

$$= y_0 + hy_0' + \frac{1}{2}h^2y_0'' + \frac{1}{6}h^3y_0''' + \frac{1}{24}h^4y_0^{(iv)} + \cdots \quad (2.7)$$

$$y_2 = y_0 + (x_2 - x_0)y_0' + \frac{1}{2}(x_2 - x_0)^2y_0'' + \frac{1}{3!}(x_2 - x_0)^3y_0''' + \frac{1}{4!}(x_2 - x_0)^4y_0^{(iv)} + \cdots$$

$$= y_0 + 2hy_0' + 2h^2y_0'' + \frac{4}{3}h^3y_0''' + \frac{2}{3}h^4y_0^{(iv)} + \cdots \quad (2.8)$$

Now, from (2.6) (2.7)and (2.8), we get:

$$\frac{h}{3}[y_0 + 4y_1 + y_2] = \frac{h}{3}[6y_0 + 6hy_0' + h^2y_0'' + 2h^3y_0''' + \frac{5}{6}h^4y_0^{(iv)} + \cdots]$$

$$= 2hy_0 + 2h^2y_0' + \frac{4}{3}h^3y_0'' + \frac{2}{3}h^4y_0''' + \frac{5}{18}h^5y_0^{(iv)} + \cdots \quad (2.9)$$

Now from (2.5) and (2.9) we obtain,

$$\int_{x_0}^{x_2} ydx - \frac{h}{3}[y_0 + 4y_1 + y_2] = \left(\frac{4}{15} - \frac{5}{18}\right)h^5y_0^{(iv)} + \cdots = -\frac{1}{90}h^5y_0^{(iv)} + \cdots$$

This is the error committed in the interval$[x_0, x_2]$.

Generally, the composite Simpson's 1/3 rule we need an even number of subdivisions. Let [a,b] be sub-divided into n even number of subdivisions, $a = x_0 < x_1 < \cdots < x_n = b$,then

From the previous results we have,

$$\int_{x_0}^{x_2} ydx = \frac{h}{3}[y_0 + 4y_1 + y_2]$$

$$\int_{x_2}^{x_4} ydx = \frac{h}{3}[y_2 + 4y_3 + y_4]$$

And similarly, we obtain

$$\int_{x_{n-2}}^{x_n} ydx = \frac{h}{3}[y_{n-2} + 4y_{n-1} + y_n]$$

Therefore, the integral over the whole interval is found by adding these integrations and is equal to

$$\int_{x_0}^{x_n} y dx = \frac{h}{3}[y_0 + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2}) + y_n].$$

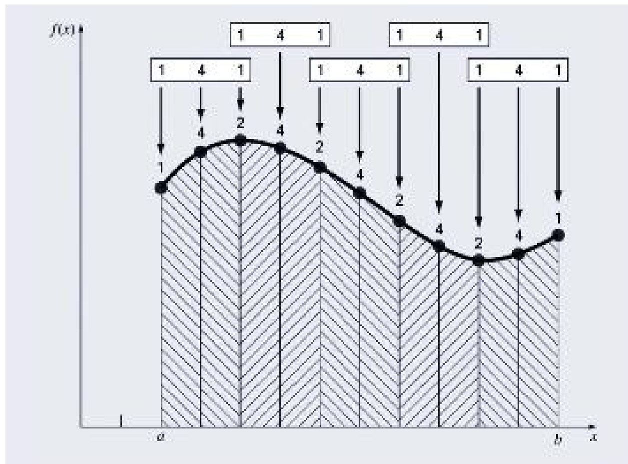

Figure 4. The composite Simpson's 1/3 rule (Levy, 2010).

In a similar manner we obtain the errors in the remaining sub-intervals $[x_2, x_4], \cdots, [x_{n-2}, x_n]$,we have:

$E = -\frac{1}{90}h^5\left[y_0^{(iv)} + y_1^{(iv)} + \cdots + y_{n-2}^{(iv)}\right]$.

$= -\frac{1}{180}(b-a)h^4 f^{(iv)}(\xi)$, $\xi \in [a,b]$where,$f^{(iv)}(\xi)$ is the largest value of the n-quantities on 4th derivatives.

2.1.3. Simpson's 3/8 –Rule

Here, we let $n = 3$ in the above equation (2.2) and obtain the following result

$$\int_0^3 f(x)dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)].$$

Using the same techniques as the above, we obtain the error committed in this method to be$E = -\frac{3}{80}h^5 f^{(iv)}(\xi)$.

This shows that the Simpson's 3/8 –rule is not as accurate as the Simpson's 1/3 –rule.

Now, we again need to derive the composite Simpson's 3/8 rule. Setting and using the formula derived so far, we obtain,

$$\int_{x_0}^{x_3} y dx = \frac{3h}{8}[y_0 + 3y_1 + 3y_2 + y_3].$$

$$\int_{x_3}^{x_6} y dx = \frac{3h}{8}[y_3 + 3y_4 + 3y_5 + y_6].$$

.
.
.

$$\int_{x_{n-3}}^{x_n} y dx = \frac{3h}{8}[y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n].$$

And then we get the integral over $[a, b]$to be the sum of all these integrals

$$\int_{x_0}^{x_n} y dx = \frac{3h}{8}[y_0 + 3y_1 + 3y_2 + 2y_3 + 3y_4 + 3y_5 + 2y_6 + \cdots + 3y_{n-2} + 3y_{n-1} + y_n].$$

And the error over the interval is obtained to be$E = -\frac{n}{80}h^5 f^{(iv)}(\xi)$.

3.1.4. Boole's and Weddle's Rules

If we wish to retain differences up to those of the fourth order we should integrate between x_0 and x_4 and obtain Boole's formulas as follows: i.e. $n = 4$

$$\int_0^4 f(x)dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)].$$

Similarly, if we integrate the interpolating polynomial between x_0 and x_6 we obtain the Weddle's formula as follows

$$\int_0^6 f(x)dx = \frac{3h}{10}[f(x_0) + 5f(x_1) + f(x_2) + 6f(x_3) + f(x_4) + 5f(x_5) + f(x_6)]$$

and the error in this method is found to be $E_6 = \frac{h^7}{140} f^{(vi)}(\xi)\ , x_0 < \xi < x_6$

4. RESULTS AND DISCUSSION

4.1. Comparison of Methods Using C++ Program

As it has been discussed in the previous sections, we have derived the formulae for the Newton-C'otes quadrature formulae and their rate of convergences. Also, we have seen that the order of convergence of Trapezoidal rule is 2 while the order of convergence of the simpsons-1/3 rule is 4. Now in this section we will use computer programs to compare and analyze the approximation methods. We will write C++ codes for each of the methods so as to compare their nature of convergence and accuracy.

For all of the programs below we have the following input output parameters;

INPUT

Accepts Number of iterations(sub divisions) and interval of integration from the user

OUTPUT

Displays Approximate value of the Integral.

```
// a simple C++ program for the Trapezoidal rule
#include<iostream.h>
#include<conio.h>
#include<math.h>
float f(float t)
{
return f(t);
}
int main()
{
floata,b,m,So,Sn,S,s=0,x,h;
int n ,i=1;
//clrscr();
cout<<"enter end points of your interval [a,b]"<<endl;
cout<<"enter the lower boundary"<<endl;
cin>>a;
cout<<"enter upper boundary"<<endl;
cin>>b;
So=f(a);
Sn=f(b);
cout<<"enter the number of partitions"<<endl;
cin>>n;
h=(b-a)/n;
do{
x=a+i*h;
s=s+f(x);
i++;
}while(i<n);
S=h*(So+Sn+2*s)/2;
cout<<"the approximate value of the integral  is ="<<S<<endl;
getch();
```

```
}
//a simple C++ programe for the simpsons1/3 rule
#include<iostream.h>
#include<conio.h>
#include<math.h>
double f(double t)
{
return f(t);
}
int main()
{
double a,b,m,So,Sn,S,s1=0,s2=0,x,h;
int n ,i=1;
//clrscr();
cout<<"enter end points of your interval [a,b]"<<endl;
cout<<"enter the lower boundary"<<endl;
cin>>a;
cout<<"enter upper boundary"<<endl;
cin>>b;
So=f(a);
Sn=f(b);
cout<<"enter the number of partitions"<<endl;
cin>>n;
h=(b-a)/n;
do{
x=a+i*h;
s1=s1+4*f(x);
i++;
x=a+i*h;
s2=s2+2*f(x);
i++;
```

```
}while(i<n);
S=h*(So+Sn+s1+s2-2*f(x))/3;
cout<<"the approximate value of the integral  is ="<<S<<endl;
getch();
}
//A C++programe for simpsos 3/8 rule
#include<iostream.h>
#include<conio.h>
#include<math.h>
double f(double t)
{
return f(t);
}
int main()
{
double a,b,m,So,Sn,S,s1=0,s2=0,s3=0,x,h;
int n ,i=1;
//clrscr();
cout<<"enter end points of your interval [a,b]"<<endl;
cout<<"enter the lower boundary"<<endl;
cin>>a;
cout<<"enter upper boundary"<<endl;
cin>>b;
So=f(a);
Sn=f(b);
cout<<"enter the number of partitions"<<endl;
cin>>n;
h=(b-a)/n;
do{
x=a+i*h;
s1=s1+3*f(x);
```

```
i++;
x=a+i*h;
s2=s2+3*f(x);
i++;
x=a+i*h;
s3=s3+2*f(x);
i++;
}while(i<n);
S=3*h*(So+Sn+s1+s2+s3-2*f(x))/8;
cout<<"the approximate value of the integral  is ="<<S<<endl;
getch();
}
```

Now, we will apply these C++ programs to evaluate different integrals and compare the methods developed above based on the outputs of the program by varying the step size and number of iterations (inputs) and finally draw some conclusions and put recommendations.

- Suppose we want to evaluate the definite integral $\int_0^1 x^4 dx$ using the methods above.

Using the code above the Trapezoidal rule with the number of partitions n=100, the value of the integral is 0.200033 while using the Simpson's1/3rule the value of the integral 0.20000 and the exact value of the integral is 0.2.

But, with n=8,the value of the integral using Simpson's1/3 rule is =0.200033 which is equal to the result with trapezoidal rule using n=100. This result shows that the Simpson's 1/3 rule converges much faster than that the trapezoidal rule, especially for polynomial integrands.

The results of the approximations (computer programs) for different integrands with different number of iterations are tabulated below in table 1.

Table 1.Outputs of the above C++ programs for polynomial integrands.

Quadrature method	***Integrand***					
	x		x^3		x^4	
	n=6	n=60	n=6	n=60	n=6	n=90
Trapezoidal rule	0.5	0.5	0.256944	0.250069	0.209234	0.200041
Simpson's 1/3 rule	0.5	0.5	0.25	0.25	0.200103	0.20000
Simpson's 3/8 rule	0.5	0.5	0.25	0.25	0.200231	0.20000
Exact value	0.5		0.25		0.2	

So,with polynomial integrands, Simpson's 1/3 rule is much more efficient than the trapezoidal and the Simpson's 3/8 rules. We also see that the methods get close to the exact value as 'n' gets larger.

- Suppose again we want to evaluate the definite integral $\int_0^1 \frac{1}{1+x}\,dx$ and $\int_0^1 e^{x^2}$ dx

Using the Trapezoidal rule, the Simpson's 1/3 rule and Simpson's 3/8 rules with n=6 and n =90, the results are tabulated below in table 2.

Table 2. Outputs of the above C++ programs for non-polynomial integrands.

Quadrature Method	***Integrand***			
	e^{x^2}		$\frac{1}{1+x}$	
	n=6	n=90	n=6	n=90
Trapezoidal rule	1.47518	1.46271	0.694877	0.693155
Simpson's 1/3 rule	1.46287	1.46265	0.69317	0.693147
Simpson's 3/8 rule	1.46313	1.46265	0.693195	0.693147
Exact value	1.46265		0.693147	

Now, from these results we observe that, with n=6, all of the methods give results with larger error. But relatively the Simpson's 1/3 and the Simpson's 3/8 rules give better results. While with n=60, the Simpson's 1/3 and the Simpson's 3/8 rules produce much more closer results, that is, exact to five decimal digits, while the Trapezoidal rule produce a result exact only to two decimal digits.

Moreover, for the approximation of the integral $\int_0^1 e^{x^2}$ dx using the trapezoidal rule, a result which is exact to five decimal digits cane obtained by taking n=365. But, this needs high number of iterations. This means the Trapezoidal rule converges slowly relative to Simpson's rules.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

Evaluating integrals using numerical methods is very useful in that, the methods/algorithms can be coded easily to some computer programs and results in an accurate solution within a short span of time. This results in obtaining fast and efficient solutions to any definite integrals, even if the function is not given explicitly but merely known only at a finite number of sample points.

From the results of the error analysis and computer programs above, it is shown that the Trapezoidal rule gives exact results independent of the number of iterations for the first degree polynomials only, while the Simpson's 1/3 rule and the Simpson's 3/8 rules gives exact result for polynomials of degree up to 3 and 4 respectively. Moreover, Simpson's1/3 rule is considerably more accurate than the Trapezoidal and the Simpson's 3/8 rules, especially for smooth integrands, it converges fast to the exact value even within very small number of subdivisions relative to the other rules. It is also observed that obtaining an approximate solution of higher accuracy using the trapezoidal needs large numbers of iterations ad hence higher effort and this was due to low order of convergence of the method.

The use and choice of the quadrature formulae discussed so far depends on the nature and type of the problem to be solved ad umber of iterations/sub divisions used. In Trapezoidal, there is no limitation; it is applicable for any number of ordinates. In Simpson's, the number of divisions should be even in number, while in Simpson's 3/8 rules it requires the number of subdivisions be multiple of 3.

5.2.Recommendations

In using the quadrature methods for applications one must be able to write and understand computer programs for the methods and in order to get better results of the study and should be able to explain what the results show. Now a day's numerical methods are becoming very popular and useful for science and engineering applications and for research studies as well. So a great attention should be given to the further studies and software developments of these methods.

Discontinuities are also a major concern and must be carefully considered for numerical methods as well. On choosing a quadrature formula for application a great care should be given on the compatibility of the method with the integrand since a method effective for one integrand may not be effective for another integrand.

6. ACKNOWLEDGEMENTS

Authors are thankful to the reviewers for their valuable suggestions and comments.

7. CONFLICT OF INTEREST

There are no conflicts of interest.

8.REFERENCE

Alomari, M. W & Dragomir, S. S. 2014. Various error estimations for several Newton- Cotes Quadrature formulae in terms of at most first derivative and applications in numerical integration. *Jordan Journal of Mathematics and Statistics*, **7(2)**: 89-108.

Amos, G & Subramanian, V. 2013. *Numerical Methods for Engineers and Scientists*. 3rdedition, Wiley, ISBN-10: 1118554930, ISBN-13: 978-1118554937, 576p.

Burden, Richard. L & Douglas, F. J. 2010.*Numerical Analysis*. 9th edition, Electronic resource, Pacific Grove, Calif Brooks, https://trove.nla.gov.au/version/264168474.

Burg, C.O.E. 2012. Derivative based closed Newton Cotes numerical quadrature. *Applied Mathematics and Computation*, **218(13)**: 7052–7065.

Chapra, Steven. C. 2006. *Applied Mathematical methods with MATLAB, for engineers and scientists*. 3rd edition, McGraw-Hill, ISBN 978-0-07-340110-2, MHID 0-07-340110-2, 653p.

Jain, M. K., Iyengar, S. R. K & Jain, R. K. 2004. *Numerical methods.* 2nd edition, ISBN: 9788122424263, New Age International Publisher, 421p.

Kharab, A & Guenther, R.B. 2001. An *Introduction to Numerical Methods–A MATLAB Approach*. CRC Press, Boca Raton, FL.

Kreyszig, E. 2006. *Advanced engineering mathematics*. 9thedition, Wiley, ISBN-13: 978-0-471-72897-9.

Levy, D. 2010. *Introduction to Numerical Analysis*. https://www.scribd.com/document/331408400/Introduction-to-Numerical-Analysis, 127p.

Mathews, J. H & Kurtis, K. F. 2004. Numerical Methods Using Matlab. 4thedition, ISBN: 0-13-065248-2, New Jersey, USA.

Rahman, Q.I & Schmeisser, G. 1990. Characterization of the speed of convergence of the trapezoidal rule. *Numerische Mathematik*, **57(1)**: 123–138.

Rao, V. Dukkipati. 2010. *Numerical methods*. ISBN 10: 8122428134, ISBN 13: 9788122428131 New Age International Publisher, 368p.

Sastry, S. S. 2012. *Introductory methods of Numerical Analysis*. 5thedition, PHI, ISBN 10: 8120345924, ISBN 13: 9788120345928, 464p.

Weidman, J. A. C. 2002. Numerical Integration of Periodic Functions: A Few Examples. *The American Mathematical Monthly*, **109(1)**: 21–36.