Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.6, No.10, 2016
www.iiste.org

Error Estimation of Numerical Integration Methods

R. Zafar Iqbal*↑ M. O. Ahmad**

*Department of Applied Mathematics, National College of Business Administration and Economics, 40-E1, Lahore-54660, Pakistan.

**Department of Mathematics and Statistics, The University of Lahore, Lahore. Pakistan

Abstract

We compare the accuracy of numerical integral methods like Newton-Cotes method and Gaussian Quadrature Rule (GQR) for the model problem and tested for another problem to verify the results. From results we notice that error of GQR is about 10 times less than Newton-Cotes formulas. For this reason we prefer GQR over other methods. But GQR uses nodes and weights which is a tedious work. This difficulty can overcome by using the idea of "three-term recurrence" relation. We can transform the problem of finding the nodes and weights for GQR to one of finding eigenvalues and eigenvectors of a symmetric tridiagonal matrix.

Keywords: *Numerical integration; Gaussian Quadrature rules; error estimate; convergence rate.*

1. Introduction

Mathematicians and scientists are sometime confronted with definite integrals which are not easily evaluated analytically, even a function $f(x)$ is known completely. To overcome this difficulty the numerical methods are used. In computational physics it is usual to evaluate integrals numerically that cannot be evaluated analytically. Numerical integration is an important section of numerical analysis. Numerical integration has many applications in science and engineering. In recent past a wide variety of novel schemes have been developed to solve integrals numerically.

Numerical integration involves replacing an integral by a sum. The term *quadrature* is used as a synonym for numerical integration in one dimension. Numerical quadrature has a long and distinguished history, including contributions by Newton, who devised the basis of what is now known as the Newton-Cotes scheme, and Gauss, who devised Gaussian quadrature [2]. The theory of numerical integration has a very long history. About 2000 years

ago, Archimedes approximated the area of circles by using inscribed and circumscribed polygons. An Arab mathematician Ibrahim, in the 10th century, introduced a more general method of numerical integration to calculate the area of the parabola. With the advent of the calculus in the 17th century, the theory of numerical integration developed rapidly [8]. In the twentieth century, numerous additional schemes were devised, including extended Simpson rules, adaptive quadrature, Romberg integration, Clenshaw-Curtis integration and others.

Accordingly, numerical integration has become an important device for processing complicated engineering designs. It is therefore important to gain a gratitude for the scope of numerical integration and its power to solve real engineering problems.

In past, slight interest has been shown to the issues of very high precision quadrature, because few serious applications have been known for such techniques, [2] and also because techniques that work well for standard machine precision often do not range well to the area of high precision. The software packages Mathematica and Maple include arbitrary precision arithmetic, together with numerical integration to high precision. These facilities are generally quite good, although in many cases they either fail or require unreasonably long run times.

Let $f(x)$ be a function which is defined on some interval [a,b] and on the set of distinct points $\{x_0, x_1, \dots, x_n\}$. Then the numerical integration for approximation can be defined as

$$\int_a^b f(x)\,dx \cong \sum_{i=0}^{n} w_i\, f(x_i) \qquad (1)$$

where w_i are the *quadrature weights* and x_i the *quadrature points*. There are a number of numerical integration methods for evaluation of definite integrals. The most commonly used methods are the Newton-Cotes formulas and Gaussian quadrature rules. Among the most popular methods for approximating the evaluation of the definite integrals are the trapezoidal rule and the Simpson rules. To improve the approximation, the interval of integration is subdivided into smaller subintervals, or segments. Increasing the number of segments results in decreasing the error until the round-off errors begin to dominate and the error begins to increase.

Here we shall give a brief introduction and implementation for these methods.

2. Methods for Numerical Integration

All most every numerical analysis textbook has a chapter on numerical integration. These present two families of quadrature rules based on (n+1) points. First one is Newton-Cotes formulas which are based on equally spaced points and the others are Gauss formulas, which are based on optimal points. A basic question to ask about any family of quadrature formulas is about its convergence [13], does it converge as $n \to \infty$, and if so, how speedy?

It has been known to all users that the Newton-Cotes formulas do not converge for a general integrand $f(x)$. It is only possible if $f(x)$ is analytic in a large region with the known interval of integration. To get improved approximation, the number of segments is increased. This is done by dividing the integration-interval into smaller subintervals. In a result error decreases until round-off errors begin to dominate and the error begins to increase [1]. On the other hand, Gauss quadrature formulas, converges for any continuous f and have no problems with rounding errors. The Newton-Cotes (n+1) points formula exactly integrates polynomials of degree n where as the (n+1) points Gauss formula exactly integrates polynomials of degree 2n + 1. Calculating nodes and weights of Gauss quadrature take some extra work, but this can be done in O(n^2) operations by solving a tridiagonal eigenvalue problem [4].

The **Newton-Cotes formulas**, the most commonly used numerical integration methods, approximate the integration of a complicated function by replacing the function with many *polynomials* [10,]. The integration of the original function can then be obtained by summing up all polynomials whose *"areas*" are calculated by the weighting coefficients and the values of the function at the nodal points [5].

The **Gaussian quadratures** give the flexibility of choosing weights and nods (abscissas) for the evaluation of the functions. Consequently, Gaussian quadratures yield twice as many places of accuracy as that of the Newton-Cotes formulas with the same numbers. For a *known* and *smooth* function, the Gaussian quadratures usually have significant advantages in efficiency. Nevertheless, data obtained from measurements in engineering problems are not always smooth or located right on the abscissas which are not uniformly spaced. Hence, the Gaussian quadratures are not apposite for such cases, as mentioned by Hildebrand (1956) in [6].

2.1. Newton-Cotes Formulas

The numerical integration methods that are derived by integrating the Newton interpolation formulas are termed as *Newton-Cotes integration formulas*. The Trapezoidal Rule and Simpson's Rules are members of this family. Here we will begin by deriving the basic trapezoidal rule and Simpson's integration formula for continues functions, on closed range.

2.2 Trapezoidal Rule

The trapezoidal rule is a numerical integration method derived by integrating the linear polynomial interpolation. It is written as

$$\mathrm{I} = \int_a^b f(x)dx = \frac{b-a}{2}[\, f(a) + f(b)] \; + E \qquad (2)$$

where E represents the error of trapezoidal rule. This error is high when we approximate the area under a curve by using single trapezoid [11]. The interval [a, b] can be divided into n intervals with equal width h. the points are $a = x_0, x_1, x_2, ..., x_n = b$ where $x_i = x_0 + ih$, for all $i = 1,2,....,n.$ The value $h = \dfrac{b-a}{n}$.

Above relation for n-interval case can be written as

$$\mathrm{I} = \int_a^b f\,(x)dx = \frac{h}{2}[\, f(a) + 2\sum_{i}^{n-1} f(a+ih) + f(b)] + E \qquad (3)$$

$$\mathrm{I} = \int_a^b f\,(x)dx = \frac{h}{2}[\, f(a) + 2\, f(x_1) + f(x_2) + ... + 2f(x_{n-1}) + f(b)] + E \qquad (4)$$

If we replace $f(a) = f_0$, $f_1 = f(a+h)$ and $f_i = f(a+ih)$ then above relation can be expressed as

$$I = \frac{h}{2}[\, f_0 + f_1 + f_2 + ... + 2f_{i-1} + f_i] + E \qquad (5)$$

The error of the trapezoidal rule is given as:

$$E = -\frac{1}{12}(b-a)\; h^2 \; f''(\xi) \qquad (6)$$

where a $\leq \xi \leq b$.

It is clear that the error of the trapezoidal rule is proportional to f'' and decreases proportionally to h^2 when we increase the number of intervals. The error is large for the

single segment trapezoidal rule. To reduce this error, we divide the interval into subintervals and then apply the trapezoidal rule over each segment.

2.3 Simpson's $\frac{1}{3}$ Rule

Simpson's $\frac{1}{3}$ rule is based on quadratic polynomial interpolation. In general the Simpson's rule is used for equally spaced data with width h. Results obtained by the trapezoidal rule lead us to think that we might be able to do better than the trapezoidal rule by using high degree polynomial [3].

In general we write:

$$\mathrm{I} = \frac{h}{3}\left[f(a) + 4\sum_{i=1,odd..i}^{n-1} f(a+ih) + 2\sum_{i=2,evev..i}^{n-2} f(a+ih) + f(b)\right] + E \qquad (7)$$

where E denote error in Simpson's rule which is obtained as:

$$E = -(b-a)\frac{h^4}{180} f^{iv}(\xi) \qquad (8)$$

where $a \leq \xi \leq b$.

For three points x_0 = a, x_1 = a + h and x_2 = b , Simpson's rule can be written as

$$I = \int_a^b f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] + E \qquad (9)$$

This error would be zero if $f(x)$ is a polynomial of degree 3 or less. Simpson's rule is easy to apply and it is considered reasonable for many applications. Error of Simpson's rule is high for n = 2 and its accuracy can be enhanced by dividing interval [a, b] into several subintervals. These intervals should be even for Simpson rule.

In general we write:

$$\mathrm{I} = \frac{h}{3}\left[f(a) + 4\sum_{i=1,odd\ i}^{n-1} f(a+ih) + 2\sum_{i=2,even\ i}^{n-2} f(a+ih) + f(b) + E\right] \qquad (10)$$

Setting $f_i = f(a+ih)$ in above relation we get

$$I = \frac{h}{3}[f_0 + 4(f_1 + f_3 + \ldots + f_{n-1}) + 2(f_2 + f_4 + \ldots + f_{n-2}) + f_n] + E \qquad (11)$$

In Simpson's rule N, the number of intervals, must be even. As the error is related to h, it varies accordingly, because the number of points $N \propto \frac{1}{h}$.

For many integrals without singularities, above mentioned Simpson's formula is perfectly adequate, and there is no need to go to higher orders. However, if high accuracy is required then higher-order formulas, like *Gaussian quadrature rules* can be used.

2.4 Gaussian Quadrature Rule

The numerical integral methods described earlier have a simple choice of points to evaluate a function $f(x)$ in a known interval [a,b]. These methods are based on equally space points. When we have freedom of choice regarding evaluation points, then more accuracy can be achieved.

Gaussian quadrature is a powerful tool for approximating integrals. The quadrature rules are all based on special values of *weights* and *abscissas*. Abscissas are commonly called evaluation points or "Gauss points", which are normally pre-computed and available in most standard mathematics tables. Algorithms and computer codes are also available to compute them.

Let the function $f(x)$ be defined on the closed interval [a, b] and that it is continuous and differentiable on this interval [12]. Further, let w be a weight function which is well defined and positive as well as continuous and integrable on [a, b]. We desire to construct quadrature formulae for the approximate evaluation of the integral

$$\int_a^b w(x)f(x)dx \qquad (12)$$

The computation of the weights and the abscissas in a Gauss quadrature rule requires little work when the system of orthogonal polynomials is already known. On the other if it is not known, at the very least it is necessary to construct the polynomial from the system with roots as the abscissas. Here we apply straightforward approach to avoid the construction of polynomial, which is easier.

The two-point Gauss quadrature rule for a function $f(x)$ can be evaluated between fixed limits a and b as follow:

$$I = \int_a^b f(x)dx = c_1 f(x_1) + c_2 f(x_2) \qquad (13)$$

There are four unknowns, c_1, c_2, x_1 and x_2 which can determined by integrating a general third order polynomial, $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ which has 4 coefficients.

2.4.1 Higher point Gaussian Quadrature Formulas

To get more accurate results, the number of Gaussian Quadratures are in creased. For this three points or higher point Gaussian Quadrature rule can be used. Three points Gaussian Quadrature rule is written as:

$$I = \int_a^b f(x)dx = c_1 f(x_1) + c_2 f(x_2) + c_3 f(x_3) \qquad (14)$$

When n points are used, we call the method an "n-point" Gaussian method, which can be used to approximate a function f(x) between fixed limits as:

$$I = \int_a^b f(x)dx \cong c_1 f(x_1) + c_2 f(x_2) + \ldots\ldots + c_n f(x_n) \qquad (15)$$

Gaussian Integration is simply based on the use of a polynomials to approximate the integrand f(t) over the interval [-1, 1] as the sum $\sum_{0\le j\le n} w_j f(x_j)$. The abscissas x_j are the roots of the Legendre polynomial $P_n(x)$ on $[-1,1]$,with degree n [?]. Whereas weights w_j are

$$w_j = \frac{-2}{(n+1)P_n'(x_j)P_{n+1}(x_j)} \qquad (16)$$

Clearly the abscissas and weights are independent of $f(x)$.

The accuracy and optimality of results depend on the choice of polynomial. The coefficients of this polynomial are unknown variables which can be determined by using any suitable method. The simplest form of Gaussian quadrature uses a uniform weighting over the interval. The particular points at which we have to evaluate $f(x)$ are the roots of a particular class of polynomials, the Legendre polynomials, over the interval. Gaussian quadrature formulae are evaluating using abscissa and weight. The choice of n is not always clear, and experimentations are useful to see the influence of choosing a different number of points.

We compute the Legendre polynomial function values using n-long iteration of the recurrence $P_0(x) = 0, P_1(x) = 1$ and for $k \ge 2$,

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x) \qquad (17)$$

The derivative is computed as

$$P_n'(x) = \frac{n\left(xP_n(x) - P_{n-1}(x)\right)}{\left(x^2 - 1\right)} \qquad (18)$$

If the integral is not posed over the interval [-1, 1] then we can apply a simple change of variable, linear scaling, to rewrite any integral over [a, b] as an integral over [-1, 1].

2.5 Error Estimation

In numerical integration, we use a finite summation to approximate the value of an integral:

$$I(f) = \int_a^b f(x)dx \cong \sum_{i=0}^{n} w_i f(x_i) + E \qquad (19)$$

There are two type of errors in numerical integration [14]. These are truncation error and round off error. Without effective evaluation of error $|I(f)-A(f)|$, a quadrature rule is of no importance. Round off error comes from the fact that we can only compute the summation to finite precision, due to the limited accuracy of a computer's representation of floating point numbers. The round off error, in general, is insignificant compared with truncation error. Conversely if the number of abscissas in the summation of relation (17) above, is immense then rounding error might need to be taken into account. This criterion is also same when the tolerance is too small [7]. As we focus on high precision integration thus ignore the round off error.

2.6 Error in Gaussian Quadrature

The error in the Gaussian quadrature rule [1] is

$$E_n = I(f)-G_n(f)=\frac{(b-a)^{2n+1}(n!)^4 f^{(2n)}(\xi_n)}{(2n+1)[(2n)!]^3} \qquad (20)$$

where a < ξ_n < b and $I(f)$ and $G_n(f)$ denote values of function calculated analytically and by using the Gaussian quadrature formula. Obviously it is computationally expensive or even difficult to evaluate $f(x)$ at many points. Consequently it is important to consider how the error E_n depends on n. Clearly one would like to get small E_n when n becomes large. There are two firmly linked ways to describe the reliance of E_n and n: (1) The order of accuracy and (2) the degree of the quadrature rule. The order of accuracy shows how fast $|E_n|$ decays to zero when n becomes large. Whereas the degree of the quadrature rule shows

for which polynomials the quadrature rule G_n is exact. A quadrature formula of degree d integrates exactly to all polynomials up to degree d.

2.7 Orthogonal Polynomials and Gaussian Quadrature

Orthogonal polynomials are classes of polynomials $p_n(x)$ defined over a range [a, b] that obey an orthogonality relation

$$\int_a^b w(x)p_m(x)p_n(x)dx = \delta_{mn}\, c_n \qquad (21)$$

where $w(x)$ is a weighting function and δ_{mn} is the Kronecker delta. Orthogonal polynomials are very useful in finding the solution of mathematical and physical problems. Such polynomials can be constructed by Gram-Schmidt orthonalization of the monomials $1, x, x^2, ...,$. Our goal is to find nodes and weights. This is a tedious work . This difficulty is overcome by using the idea of "three-term recurrence". We will develop this theory a little in this section.

Theorem 1. The abscissas of the N -point Gaussian quadrature formula are precisely the roots of the orthogonal polynomial [3] for the same interval and weighting function.

Let p_n be a nontrivial polynomial of degree n such that

$$\int_a^a w(x)\, x^k\, p_n(x)dx = 0, \quad for \quad k = 0, 1, 2, \ldots . n-1. \qquad (22)$$

If we pick the nodes to be the zeros of p_n , then there exist weights w_i which make the computed integral exact for all polynomials of degree 2n-1 or less. Furthermore, all these nodes will lie in the open interval (a, b).

For more understanding we present another theorem.

Theorem 2. Construction of Gaussian quadrature

For N = 2n-1, there exists [3] a set of Gaussian points $x_i^{(n)}$ and weights $w_i^{(n)}$ such that

$$\int_{-1}^{1} x^k dx = \sum_{i=1}^{n} w_i^{(n)} \left(x_i^{(n)}\right)^k \qquad (23)$$

hold for all k = 0,1,2,…,N

3.0. Comparison of Newton-Cotes rules with Gaussian quadrature rule

In this section we compare the accuracy of Newton-Cotes method and Gaussian quadrature for the model problem.

Example 1. Evaluate the integral

$$I = \int_0^1 Sin(x)dx \qquad (24)$$

by using the Trapezoidal rule , Simpson rule and Gaussian quadrature formula.

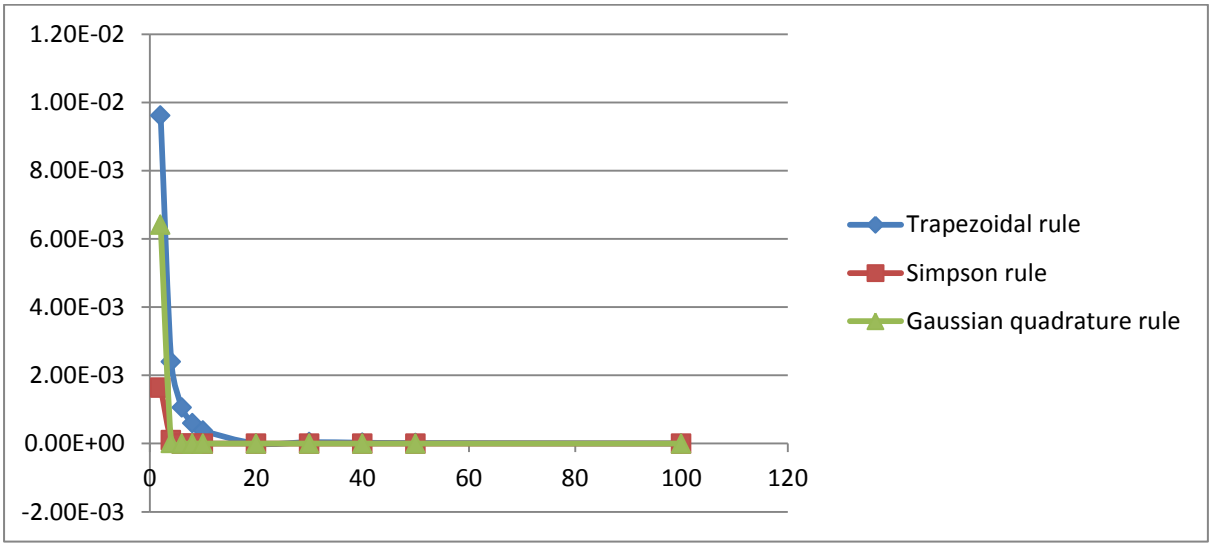
Exact value (analytical solution) is calculated by simple integration rules which is 4.5970e-001.

We will present a table about the comparisons of errors due to the Trapezoidal rule, Simpson's rule and Gaussian quadrature rule by C + + programming.

Table 1.1: Comparison of Errors

N- values	Trapezoidal rule	Simpson rule	Gaussian quadrature rule
2	9.61E-03	-1.65E-03	6.42E-03
4	2.40E-03	-1.01E-04	1.56E-05
6	1.06E-03	-1.98E-06	5.23E-10
8	5.99E-04	-6.25E-07	4.89E-15
10	3.83E-04	-2.56E-07	2.22E-16
20	9.58E-05	-1.60E-08	4.44E-16
30	4.26E-05	-3.15E-09	6.66E-16
40	2.39E-05	-3.15E-09	1.33E-15
50	1.53E-05	-4.09E-10	1.58E-11
100	3.83E-06	-2.55E-11	1.33E-15

Graphical representation of error comparison.



Graph-1

From the table and graph as, we see that the error of 4 points Gaussian quadrature rule is almost equal to 50 points of the Trapezoidal rule. Similarly absolute error due to using 6 points Gaussian quadrature rule is almost equal to 50 points Simpson's rule. From these results it is clear that error of Gaussian quadrature rule is about 10 times less than Newton-Cotes formulas. Further when n > 10 the error due to Gaussian quadrature becomes negligibly small.

We test our results for irrational integral which can be found in [15].

Example 2. Evaluate the integral $I = \int_0^1 \sqrt{1-x^2}\,dx$ by using the Trapezoidal rule , Simpson rule and Gaussian quadrature formula.

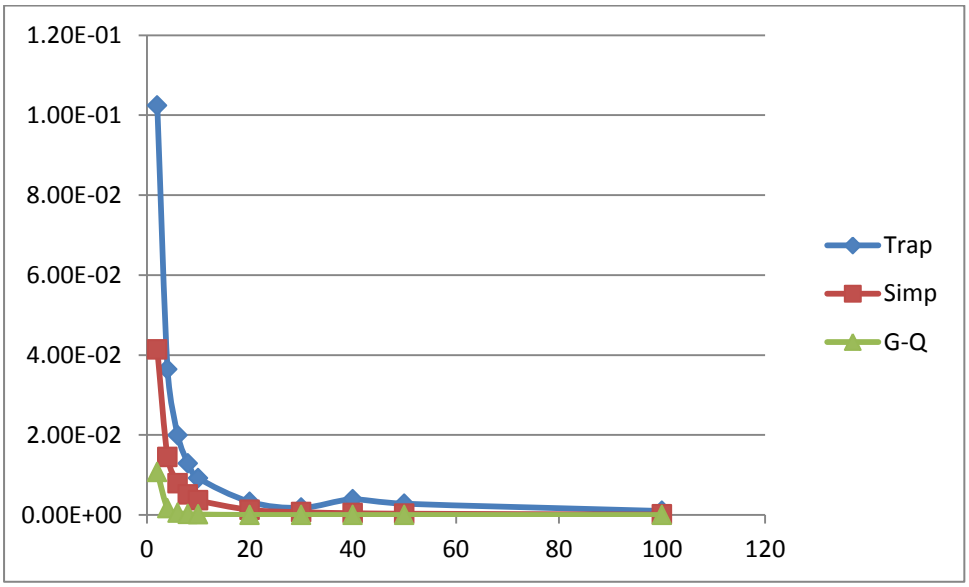
Exact value, or analytical solution, is calculated by simple integration rules which is

$\frac{\pi}{4} = 0.78539816$

Table 1.2: Comparison of Errors

N- values	Trapezoidal rule	Simpson rule	Gaussian quadrature rule
2	1.02E-01	4.14E-02	1.07E-02
4	3.64E-02	1.45E-02	1.66E-03
6	1.99E-02	7.87E-03	5.40E-04
8	1.29E-02	5.10E-03	2.40E-04
10	9.21E-03	3.65E-03	1.27E-04
20	3.28E-03	1.28E-03	1.70E-05
30	1.79E-03	7.00E-04	5.00E-06
40	3.94E-03	4.54E-04	2.00E-06
50	2.82E-03	3.25E-04	1.00E-06
100	9.99E-04	1.15E-04	0.00E+00

Graphical representation of error comparison.



Graph-2

From the table and graph as, it is clear that error due to Gaussian quadrature rule is very small as compared to Newton-Cotes formulas. Further when n > 20 the error due to Gaussian quadrature becomes negligibly small.

The main benefit of Gaussian quadrature is that of its very high-order accuracy with very few number of intervals, especially when we are using points less than 10. This accuracy

is further enhanced by increasing number of points. Due to these reasons we prefer Gaussian quadrature rule over other methods.

The computational cost of abscissa and weight by using Gaussian quadrature scheme increases quadratically with n, since each Legendre polynomial evaluation requires n steps. In general, Gaussian quadrature achieves quadratic convergence for well- behaved integrand functions. It means by doubling n achieves roughly twice as many correct digits in the quadrature result, after a few initial levels.

While the value of n required to achieve a given precision level usually increases linearly with the precision level [15], eventually the total run-time cost of computing the abscissas and weights increases even faster than n^2.

According to David H. Bailey and J.M. Borwein in [15], there is no known scheme for generating Gaussian abscissa-weight pairs that evade this quadratic dependence on n, however the computed High-precision abscissas and weights can be stored and used latter.

In future we shall use the connection between Gaussian quadrature rule and eigenvalues problem. By using three- term recurrence relation [9], and we shall construct tridiagonal symmetric matrix to compute nodes and weights for Gaussian quadrature rule.

4.0 Acknowledgements

Author is thankful to Prof Dr Roy Mathias, University of Birmingham ,U.K, for his valuable advices and guidance.

References

[1] Mohamad Adnan AI-Alaoui, *A class of numerical integration rules with first order derivatives*, ACM 31 (1996), 25-44.
[2] David H. Bailey, *A comparison of three high-precision quadrature schemes, Experimental Mathematics* 14 (2005), 317-329.
[3] James F. Epperson, An introduction to numerical methods and analysis, John Wiley and Sons,Inc, 2002.
[4] G. H. Golub and J. H. Welsch, Calculation of gauss quadrature rules, Math. Comp 23 (1969), 221-230.
[5] Nicholas Hale and Alex Townsend, *Fast and accurate computation of gauss-legendre and gauss-Jacobi quadrature nodes and weights*, SIAM J. SCI. COMPUT 35 (2013), A652{A674.
[6] F. B Hildebrand, Introduction to numerical analysis, McGraw-Hill, New York, 1956.
[7] Philip J.Davis and Philip Rabinowitz, *Methods of numerical integration*, second edition, Academic Press, 1984.
[8] Arnold R. Krommer and Christoph W. Ueberhuber, *Computational integration*, SIAM (1998).

[9] David Bau III Lloyd N. Trefethen, *Numerical linear algebra, Society for Industrial and Applied Mathematics*, 1997.
[10] Adriana Szekeres MAndreea Dragomir, *A comparative study of parallel algorithms for numerical integration*, CSMR 1 (2010).
[11] Madhumangal Pal, *Numerical analysis for scientists and engineers*, Alpha Science International Ltd, 2007.
[12] Endre Suli and David F. Mayers, *An introduction to numerical analysis*, Cambridge University Press, Cambridge, United Kingdom, 2003.
[13] Lloyd N. Trefethen, *Is gauss quadrature better than Clenshaw-Curtis*?, SIAM J. SCI. COMPUT 50 (2008), 67{87.
[14] Lingyun Ye, Numerical quadrature : *Theory and computation, Master's thesis*, Dalhousie University Halifax, 2006.
[15] David H. Bailey and J.M. Borwein, *High-precision numerical integration: Progress and challenges*, Experimental Mathematics (2009)