Annals of Pure and Applied Mathematics
Vol. 8, No. 2, 2014, 27-36
ISSN: 2279-087X (P), 2279-0888(online)
Published on 17 December 2014
www.researchmathsci.org

Annals of
Pure and Applied Mathematics

A Study on Numerical Stability of Finite Difference Formulae for Numerical Differentiation and Integration

***M. Ramesh Kumar*[1] and *G. Uthra*[2]**

[1] Department of Mathematics, Saveetha Engineering College, Thandalam
Chennai-602105. India. Email: ramjan_80@yahoo.com

[2] PG & Research Department of Mathematics, Pachaiyappa's College
Chennai- 600030, India. Email: uthragopalsamy@yahoo.com

Received 3 September 2014; accepted 21 November 2014

Abstract. The numerical differentiation based on the interpolating polynomial is basically an unstable process and one cannot expect good accuracy even when the original data are known to be accurate. We analyze the stability of computation of derivatives through polynomial interpolation at given point numerically and prove it has poor stability when closer to the interpolating nodes however it has a quite good stability between interpolating nodes.

The numerical integration by use of lower order formulas such as trapezoidal rule and Simpson rule gives accuracy of results than use of higher order Newton-Cotes formulae. In this paper, we also analyze the reason for the poor stability of higher order Newton-Cotes formulae. Numerical examples are given to study roundfoff error analysis of numerical differentiation and integration.

Keywords: Round off errors, numerical differentiation, integration, Newton-Cotes formulae, numerical stability

AMS Mathematics Subject Classification (2010): 65D25, 65D30

1. Introduction

Numerical approximations to derivatives are used mainly in two ways. First, we are interested in calculating derivatives of given data that are often obtained empirically. Second, numerical differentiation formulae are used in deriving numerical methods for solving ordinary and partial differential equations. The problem of numerical differentiation of noisy data is ill-posed, small changes of the data may result in large changes of the derivative. There is always a conflicting relationship, as nodes become denser, data reflect the rapid variation better while differentiation of the data gets more noise. Consider the central difference formula for approximating $f'(a)$

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} - \frac{h^2 f'''(\xi)}{6} \tag{1}$$

In calculations, we will in fact use the numbers, $f(a+h) + \Delta f_1$ and $f(a-h) + \Delta f_{-1}$ Instead of the numbers $f(a+h)$ and $(a-h)$, use of roundoff. Therefore, we compute

$$f'_{\text{comp}} = \frac{f(a+h)-f(a-h)}{2h} + \frac{\Delta f_1 - \Delta f_{-1}}{2\text{h}} - \frac{h^2 f'''(\xi)}{6}$$

Hence

$$f'(a) = f'_{\text{comp}} - \frac{\Delta f_1 - \Delta f_{-1}}{2h} - \frac{h^2 f'''(\xi)}{6} \quad (2)$$

The error in the computed approximation of f'_{comp} and $f'(a)$ is therefore seem to consist of two parts, one part due to roundoff, and the other part due to discretization. If $f'''(x)$ is bounded, then the discretization error goes to zero as $h \to 0$, but the round off error goes if we assume that $\Delta f_1 - \Delta f_{-1}$ does not decrease. Hence, $f'(a)$ gives good approximation only at optimum value of h. If h is small and approaches to zero i.e when the data is dense then the process estimating derivatives for evenly spaced nodes is unsatable. This analysis shows that we can combat the round off error by using "sufficiently" high precision arithmetic. But this is impossible when $f(x)$ is only approximately at finitely many points [1]. However, in Ref[5] shows that the use of a higher-order formula, such as a 7-or even a 10-point approximation, based on the method of undetermined coefficients, can sometimes lead to better accuracy and enhanced computational efficiency rather than 2- point and three point formula. In the present study we give roundoff error analysis for higher order numerical differentiation formula through polynomials for arbitrary spaced grids. In this paper, we give the round of errors of calculation of derivatives through polynomial interpolation and analyze the stability of numerical differentiation up to higher order.

The idea of numerical integration is to replace a complicated function or tabulated data with an approximating function that is easy to integrate. Polynomial function is the best choice to replace the actual function because of its simple form and also it can be easily found through Lagrange interpolation formula or Newton interpolation formula[4] for evenly or unevenly spaced grids with any degree of accuracy. If the nodes are spaced evenly then the quadrature formula is called Newton-Cotes formula. Trapezoid, Simpson's 1/3 and 3/8 rules, Bode's are special cases of 1st, 2nd, 3rd and 4th order polynomials are used, respectively in Newton cotes formulas. Using large number of equally spaced nodes may be inaccurate behavior associated with high-degree polynomial interpolation Indeed, every n-point Newton-Cotes rule with n ≥ 11 has at least one negative weight, so Newton-Cotes rules become arbitrarily ill-conditioned. The lower order formulas for approximating integrals such as Trepezoidal and Simson rules are special cases of Newton-Cotes integration formulas and gives better accuracy then higher order formulas. In this paper we analyse poor stability of higher order Newton-Cotes formula through roundoff error analysis.

2. Preliminaries

Let Π_n denote the vector space of all polynomials of degree at most n and let $x_i, i = 0, \ldots, n$, be $n+1$ distinct nodes and suppose that $f_i, i = 0, \ldots, n,$, are corresponding numbers. Then, there exists a unique polynomial $P_f \epsilon\, \Pi_{\text{n}}$ such that $P_f(x) = f_i$, $i = 0, \ldots, n$. Let $X = \{x_0 x_1, x_2, \ldots, x_n\}$ and denote the Lagrangian polynomials as follows

$$l_X(x) = \prod_{k=0}^{n}(x - x_k) \text{ and } l_{i,X}(x) = \prod_{\substack{k=0,\\ i\neq k}}^{n} \frac{x-x_k}{x_i - x_k}. \tag{3}$$

The Lagrange's interpolation formula[1, 3] for approximating f on the X is given by

$$P_{f,X}(x) = \sum_{i=0}^{n} l_{X,i}(x) f_i. \tag{4}$$

Differentiating m times with respect to x,

$$P_{f,X}^{(m)}(x) = \sum_{i=0}^{n} l_{i,X}^{(m)}(x) f_i, \tag{5}$$

where $P_{f,X}^{(m)}(x)$ and $l_{i,X}^{(m)}(x)$ are m^{th} derivatives of $P_{f,X}(x)$ and $l_{X,i}(x)$ respectively. Let $J = \{j_0, j_1, j_2, \ldots, j_n\}$ and denote the Lagrangian polynomials as follows

$$l_J(s) = \prod_{k=0}^{n}(s - j_k) \text{ and } l_{i,J}(x) = \prod_{\substack{k=0,\\ i\neq k}}^{n} \frac{s-j_k}{j_i - j_k}. \tag{6}$$

The Lagrange's interpolation for approximating f on ser J is given by

$$P_{f,J}(s) = \sum_{i=0}^{n} l_{i,J} f_i. \tag{7}$$

Definition 2.1. Define the condition number of m^{th} order derivative of $P_{f,X}(x)$ at x for $n \in NU\{0\}$ over the set X as follows[3]

$$\mathrm{cond_m}(X, x, f) = \sup_{\mathrm{x}\in \mathrm{X}} \lim_{\square\to 0} \frac{\left|P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x)\right|}{\left|P_{f,X}^{(m)}(x)\right|} \tag{8}$$

Similarly, condition number of m^{th} order derivative of $P_{f,J}(x)$ at s over the set J as follows

$$\mathrm{cond_m}(J, s, f) = \sup_{s\in J} \lim_{\square\to 0} \frac{\left|P_{f,J}^{(m)}(s) - P_{f+\Delta f,J}^{(m)}(s)\right|}{\left|P_{f,J}^{(m)}(s)\right|} \tag{9}$$

3. Round of error analysis of numerical differentiation

Define $h = \min_{\mathrm{i}}\{|x_i - x|\} \neq 0$ for $i = 0, \ldots, n$ and arrange each node x_i, $i = 0, \ldots, n$ by spacing at distance $j_i h$ from x_0, (i.e) $x_i = x_0 + j_i h$, where $j_0 = 0, j_n = (x_n - x_0)/h$ and satisfies $0 \leq j_i \leq (x_n - x_0)/h$. Let x be any point on $[x_0, x_n]$ and $x = x_0 + sh$, where $0 \leq s \leq (x_n - x_0)/h$. Then, the following relationship holds between Lagrange's polynomials (3)

$$l_X(x) = h^{n+1} l_J(s) \tag{10}$$

and

$$l_{i,X}(x) = l_{i,J}(x) \qquad (11)$$

Using (3), (6) and (8) gives

$$P_{f,X}(x) = \sum_{i=0}^{n} l_{i,X}(x) f_i = \sum_{i=0}^{n} l_{i,J}(s) f_i$$

Thus, we find that

$$P_{f,X}(x) = P_{f,J}(s) \qquad (12)$$

Equation (12) shows that $P_{f,X}(x)$ does not depend on h. Differentiating (11) with respect to x and using $ds = dx/h$, gives

$$\frac{d}{dx}\left[l_{i,X}(x)\right] = \frac{d}{dx}\left[l_{i,J}(s)\right]$$

$$= \frac{d}{ds}\left[l_{i,J}(s)\right] \times \frac{ds}{dx}$$

$$= \frac{1}{h}\frac{d}{ds}\left[l_{i,J}(s)\right]$$

$$= \frac{d}{ds}\left[l_{i,J}(s)\right] \times \frac{ds}{ds}$$

$$l_{i,X}^{(1)}(x) = \frac{1}{h} l_{i,J}^{(1)}(s)$$

Differentiating again with respect to x, gives

$$l_{i,X}^{(2)}(x) = \frac{1}{h^2} l_{i,J}^{(2)}(s)$$

Proceeding this m times, yields

$$l_{i,X}^{(m)}(x) = \frac{1}{h^m} l_{i,J}^{(m)}(s) \qquad (13)$$

Substituting (13) in (5), gives

$$P_{f,X}^{(m)}(x) = \frac{1}{h^m} \sum_{i=0}^{n} l_{i,J}^{(m)}(s) f_i \qquad (14)$$

Let Δf is the small change in f. Then

$$P_{f+\Delta f,X}^{(m)}(x) = \frac{1}{h^m} \sum_{i=0}^{n} l_{i,J}^{(m)}(s)(f_i + \Delta f_i) \qquad (15)$$

(14)-(5) $\Longrightarrow$

$$P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x) = \frac{1}{h^m} \sum_{i=0}^{n} l_{i,J}^{(m)}(s) \Delta f_i$$

$$\left|P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x)\right| \leq \frac{1}{h^m}\sum_{i=0}^{n}\left|l_{i,J}^{(m)}(s)\Delta f_i\right|$$

Choose $|\Delta f| \geq \epsilon$, gives

$$\left|P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x)\right| \leq \frac{\epsilon}{h^m}\sum_{i=0}^{n}\left|l_{i,J}^{(m)}(s)f_i\right| \tag{16}$$

Equality is attained for $\Delta f_i = \mathrm{sign}(\epsilon f_i\, l_{i,J}^{(m)}(j)f_i)$

$$\left|P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x)\right| = \frac{1}{h^m}\sum_{i=0}^{n} l_{i,J}^{(m)}(s)f_i \tag{17}$$

Similarly, we easily find that

$$\left|P_{f,J}^{(m)}(s) - P_{f+\Delta f,J}^{(m)}(s)\right| = \sum_{i=0}^{n} l_{i,J}^{(m)}(s)f_i \tag{18}$$

Using (17) and (18) , gives

$$\frac{\left|P_{f,J}^{(m)}(s)-P_{f+\Delta f,J}^{(m)}(s)\right|}{\left|P_{f,J}^{(m)}(s)\right|} = \frac{\left|P_{f,X}^{(m)}(x)-P_{f+\Delta f,X}^{(m)}(x)\right|}{\left|P_{f,X}^{(m)}(x)\right|}$$

Hence,

$$\mathrm{cond_m}(X,x,f) = \mathrm{cond_m}(J,s,f) \tag{19}$$

Hence the condition number on *X* at *x* is same as condition number *J* at *s*

Using (16) and (19), yields the bound for roundoff error of $P_{f,X}^{(m)}(x)$

$$\left|P_{f,X}^{(m)}(x) - P_{f+\Delta f,X}^{(m)}(x)\right| \leq \frac{\epsilon}{h^m}\,\mathrm{cond_m}(J,s,f)\left|P_{f,J}^{(m)}(s)\right| \tag{20}$$

Suppose that f be a real-valued function and continuously differentiable function on the closed interval $[a,b]$, $where\, a = \min\{x_0, \ldots x_n\}$ and $b = \max\{x_0, \ldots x_n\}$. Shadrin [8] has shown that if $P_{f,X}(x)$ denotes the polynomial of degree *n* interpolating f at the points $x_0, x_1, \ldots x_n$ then for $m = 0, 1, \ldots, n$

$$\left|P_{f,X}^{(m)}(x) - f^{(m)}(x)\right| \leq \left\|l_X^{(m)}(x)\right\|\left\|\frac{f^{(m+1)}}{(m+1)!}\right\|. \tag{21}$$

This bound was earlier conjectured by Howell [3] who also proved it for the highest derivative $k = n$. Let $P_{c,X}^{(m)}(x)$ is computed $P_{f,X}^{(m)}(x)$. Then

$$\left|P_{c,X}^{(m)}(x) - f^{(m)}(x)\right| \leq \left|P_{c,X}^{(m)}(x) - P_{f,X}^{(m)}(x)\right| + \left|P_{f,X}^{(m)}(x) - f^{(m)}(x)\right|. \tag{22}$$

Using (21) and (22)

$$\left|P_{c,X}^{(m)}(x) - f^{(m)}(x)\right| \leq \frac{\epsilon}{h^m} \operatorname{cond}_{\mathrm{m}}(J,s,f)\left|P_{f,J}^{(m)}(s)\right| + \left\|l_X^{(m)}(x)\right\| \left\|\frac{f^{(m+1)}}{(m+1)!}\right\|.$$

Using (13), we find the bound for total error

$$\left|P_{c,X}^{(m)}(x) - f^{(m)}(x)\right| \leq \frac{\epsilon}{h^m} \operatorname{cond}_{\mathrm{m}}(J,s,f)\left|P_{f,J}^{(m)}(s)\right| + h^{n+1-m}\left\|l_J^{(m)}(s)\right\| \left\|\frac{f^{(m+1)}}{(m+1)!}\right\|. \tag{23}$$

3.1. Numerical experiment

We report an experiment whose purpose is to verify the conclusions of the roundoff error analysis. There are plenty of numerical differentiation formulas in literature. Here, we use following formula from [6] to compute m^{th} order derivative on various distribution.

$$\frac{f^{(m)}(x)}{m!} = \frac{1}{\delta_0}\sum_{i=0}^{n} f_i w_{i,X} \sum_{k=0}^{m-\chi} \frac{a_k}{(x_i - x)^{m+1-k}} + \chi a_m f(x) + E(x). \tag{24}$$

Where $a_0 = 1$, $\sum_{k=0}^{m} a_k\, \delta_{m-k} = 0$, $\chi = \begin{cases} 1, & x = x_i \\ 0, & x \neq x_i \end{cases}$, $\delta_r = \sum_{i=0}^{n} \frac{1}{(x_i - x)^{r+1}}$ and $E(x) = l_X(x)\sum_{k=0}^{m-\chi} \frac{f^{(n+1-k+m}(\xi_k)}{(n+1-k+m)!}$. The computations were performed in MATLAB, for which $u \approx 10^{-16}$. In first example, we take 11 equally spaced points x_j on $[-1,1]$ thus $n = 10$ and set $f(x) = \mathrm{e}^{\mathrm{x}}$. We evaluate the derivatives approximately up to order 3 at 100 equally spaced points. Figure 1 and Figure 2 plots the errors for derivatives up to order 3 for 11 points on evenly spaced points and Chebyshev points of first kind.

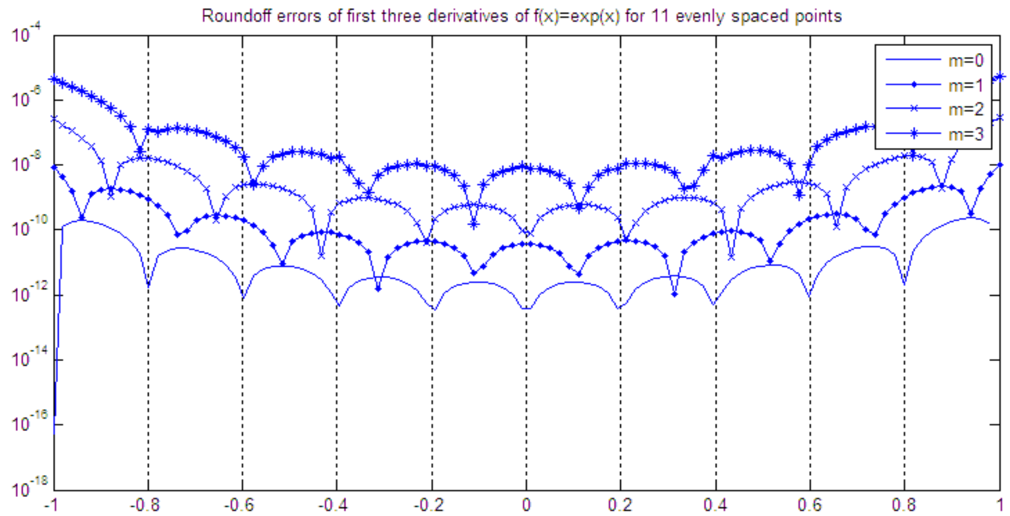

Figure 1:

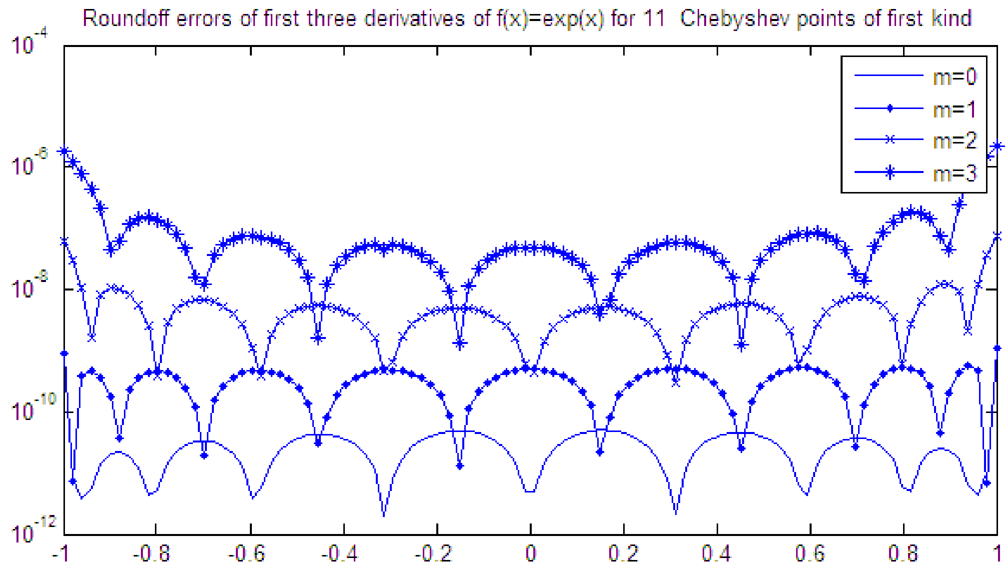

Figure 2:

In the second example, we take 11 equally spaced points x_j on $[-1,1]$ thus $n = 10$ and set $f(x) = \sin\left(\frac{\pi}{2}x\right)$. We evaluate the derivatives approximately up to order 3 at 100 equally spaced points. Figure 3 and Figure 4 plots the errors for derivatives up to order 3 for 11 points on evenly spaced points and Chebyshev points of first kind.

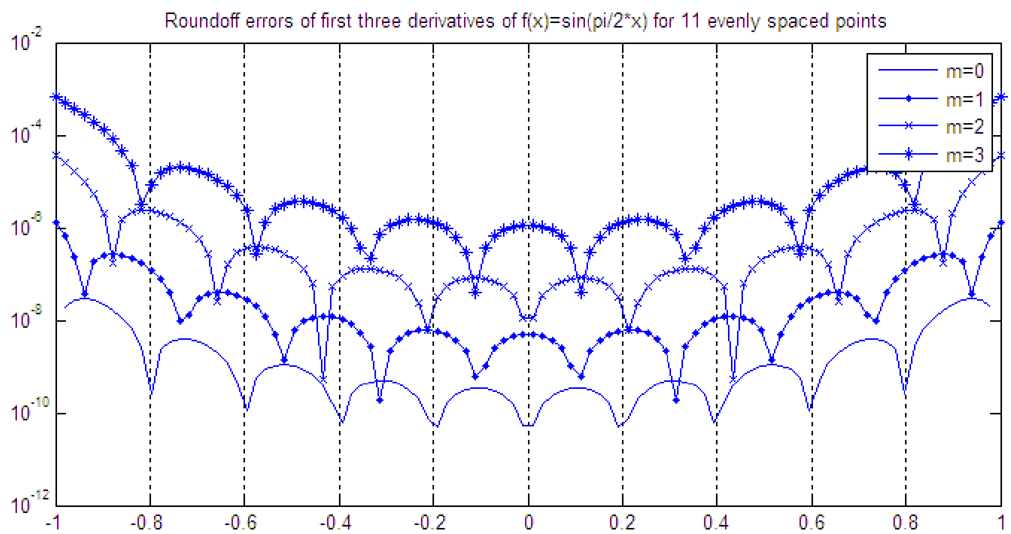

Figure 3:

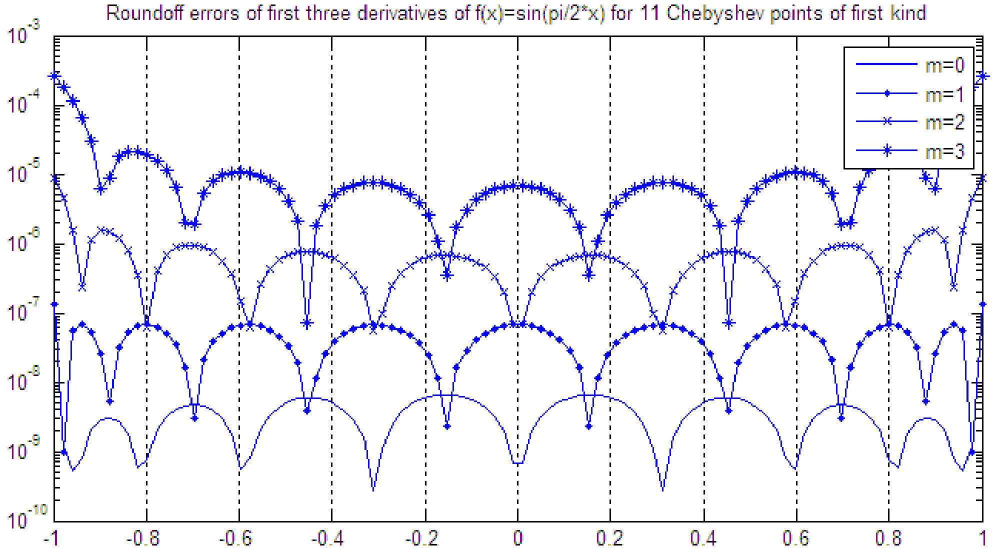

Figure 4:

The equation (23) clears that computation of derivatives through finite difference formula depends upon condition number of order of derivatives and h. These are the two factors determines stability of the computation of the numerical derivatives. Usually, if the order of derivative increases then the power of h decreses in (20). This proves that the round off error increases when the order differentiation also increases. In the first example the condition number of first three derivatives of e^x on evenly spaced points are $2982.4,\ 79925,\ 1.2861 \times 10^6$. Hence the roundoff errors of higher order derivatives increases when condition number increases. In the second example the condition number of first three derivatives on evenly spaced nodes are 3.8144×10^8, 4955.4, 2.816×10^8. But power of h decreases when the order of differentiation increases. If h is too small the roundoff error becomes very high. Hence, the calculation of derivatives near the nodes gives very poor accuracy. For larger values of h, the discretization error in (23) becomes high. Therefore, for very few points between the nodes gives quite good accuracy.

4. Roundoff error analysis of numerical integration

Let $x_0 x_1, x_2, \dots, x_n$ are distinct numbers on the closed interval $[x, x+H]$ and $f \in C^{(n+1)}[x, x+H]$. The problem of numerical integration is to approximate the definite integral $\int_x^{x+H} f(t)dt$. Since polynomials are easy to integrate by using Taylor series, we find that

$$\int_x^{x+H} P_{f,X}(t)dt = H \sum_{k=0}^{n} \frac{H^k}{(k+1)!} {P_{f,X}}^{(k)}(x). \tag{25}$$

Let Δf is the small change in f. Then

$$\int_x^{x+H} P_{f,X}(t)dt - \int_x^{x+H} P_{f+\Delta f,X}(t)dt = H \sum_{k=0}^{n} \frac{H^k}{(k+1)!} \left[{P_{f,X}}^{(k)}(x) - {P_{f+\Delta f,X}}^{(k)}(x)\right].$$

Thus, the roundoff error of numerical integration can be found as follows

$$\left|\int_x^{x+H} P_{f,X}(t)dt - \int_x^{x+H} P_{f+\Delta f,X}(t)dt\right| \le H\sum_{k=0}^{n}\frac{H^k}{(k+1)!}\left|P_{f,X}{}^{(k)}(x) - P_{f+\Delta f,X}{}^{(k)}(x)\right|$$
$$= H\sum_{k=0}^{n}\left(\frac{H}{h}\right)^k\frac{1}{(k+1)!}\left|P_{f,J}{}^{(k)}(s) - P_{f+\Delta f,J}{}^{(k)}(s)\right|.$$

Using (20), yields that

$$\left|\int_x^{x+H} P_{f,X}(t)dt - \int_x^{x+H} P_{f+\Delta f,X}(t)dt\right| \le \epsilon H\sum_{k=0}^{n}\left(\frac{H}{h}\right)^k \mathrm{cond_k}(X,x,f)\frac{|P_{f,J}{}^{(k)}(s)|}{(k+1)!} \qquad (26)$$

Newton-Cotes formulae. Let $x_i = a + ih, i = 0,1,23,\ldots,n,\ h \neq 0$ are equally spaced grids. Now replacing x by a and H by nh in and after simplification, we obtain

$$\left|\int_a^{a+nh} P_{f,X}(t)dt - \int_a^{a+nh} P_{f+\Delta f,X}(t)dt\right| \le \epsilon h\sum_{k=0}^{n} n^{k+1}\mathrm{cond_k}(X,a,f)\frac{|P_{f,J}{}^{(k)}(j)|}{(k+1)!} \qquad (27)$$

The equation gives roundoff error of $(n+1)$ *−point formula for Newton-Cotes closed integration formula.* Since the open quadrature formula do not require functional value at the limit points of integration, assume that $x_i = a + ih, i = 1,2,\ldots,n\text{-}1,\ h \neq 0$. Then

$$\left|\int_a^{a+nh} P_{f,X}(t)dt - \int_a^{a+nh} P_{f+\Delta f,X}(t)dt\right| \le \epsilon h\sum_{k=0}^{n-2} n^{k+1}\mathrm{cond_k}(X,a,f)\frac{|P_{f,J}{}^{(k)}(j)|}{(k+1)!} \qquad (28)$$

Linear multistep methods. Let $x_n, x_{n-1}, x_{n-2},\ldots,x_{n-p}$ are $p+1$ distinct numbers on the interval $[x_n, x_{n+1}]$, where $x_{n+1} = x_n + h'$. If we approximate the differential equation $y' = f(x,y)$ by integrating from x_n to x_{n+1}, gives

$$\left|\int_{x_n}^{x_n+h'} P_{f,X}(t)dt - \int_{x_n}^{x_n+h'} P_{f+\Delta f,X}(t)dt\right| \le \epsilon h'\sum_{k=0}^{n}\left(\frac{h'}{h}\right)^k \mathrm{cond_k}(X,x,f)\frac{|P_{f,J}{}^{(k)}(j)|}{(k+1)!}$$

The equation gives roundoff error of $(p+1)-$ *point predictor formula.* If we approximate the differential equation $y' = f(x,y)$ by integrating from x_{n+1} to x_n, gives

$$\left|\int_{x_n+h'}^{x_n} P_{f,X}(t)dt - \int_{x_n+h'}^{x_n} P_{f+\Delta f,X}(t)dt\right| \le \epsilon h'\sum_{k=0}^{n}\left(\frac{h'}{h}\right)^k \mathrm{cond_k}(X,x,f)\frac{|P_{f,J}{}^{(k)}(j)|}{(k+1)!}$$

This kind of multistep formula is known as $(p+1)$ *−point corrector formula.*

4.1. Numerical example

We give numerical example for roundoff errors of evaluation of integrals $\int_0^1 e^x dx$ and $\int_{-1}^{1}\frac{dx}{1+x^2}$ using *n*-point Newton-Cotes formula. The roundoff errors of number of noes $n = 2,3,\ldots,17$ are given in table 1. Trepzoidal, Simpson's 1/3 and Simson's 3/8 rule are special cases of Newton-Cotes formulas for $n = 2,3$ and 4. The table 1 shows numerical instability of higher order Newton-Cotes formulas for computation of $\int_0^1 e^x dx$ and $\int_{-1}^{1}\frac{dx}{1+x^2}$.

The equation (27) shows that the roundoff error of Newton-Cotes formula depends on all the condition number up to highest derivative n (derivative of n degree polynomial is n!) and the highest power of n in the last term n^{n+1}. Thus, n^{n+1} grows exponentially as n increases. Since all the summation in (17) are positive terms, the upper bound in (27) is very high. There is no evidence of strong stability of higher Newton-Cotes formula.

No. of nodes	Roundoff errors of $\int_0^1 e^x dx$	Roundoff errors of $\int_{-1}^{1} \frac{dx}{1+x^2}$
2	1.4086e-001	5.7080e-001
3	5.7932e-004	9.5870e-002
4	2.5832e-004	2.9204e-002
5	8.5947e-007	1.0796e-002
6	4.8452e-007	5.1855e-003
7	1.1065e-009	2.2440e-003
8	1.7479e-009	1.1900e-003
9	4.3057e-007	5.6582e-004
1l	4.8493e-007	3.0523e-004
11	4.8434e-004	6.0862e-005
12	7.2006e-003	3.6071e-003
13	5.4590e-001	2.2450e-001
14	2.9756e+001	1.9330e+001
15	2.9826e+003	3.0918e+003
16	1.2429e+006	7.8947e+004
17	1.9051e+006	2.8648e+005

Table 1: Roundoff errors for $\int_0^1 e^x dx$ and $\int_{-1}^{1} \frac{dx}{1+x^2}$ using Newton-Cotes formulas for n=2,3,…,17

5. Conclusion

In conclusion, we note that roundoff error analysis numerical differentiation and integration through polynomial interpolation have been studied in this article. It is shown that the round off errors are depend on condition number of the derivative and $h = \min_i\{|x_i - x|\} \neq 0$ for $i = 0,\dots,n$. Hence, it is clear that the computation of derivatives very close to the given nodes, posses poor numerical stability than the near the center of nodes. Similarly, we study the roundoff errors of numerical differentiation depends upon condition of umber of derivatives up to higher order and the ratio $\left(\frac{H}{h}\right)^n$. It is shown that the use of higher order Newton-Cotes formulae for interation is unstable process since the number $\frac{n^{n+1}}{n+1}$ grows exponentially as *n* increases. Therefore the use of lower order formulae such as composite trapezoidal rule and simson rule good choice then the use of higher order Newton-Cotes formula.

REFERENCES

1. S.D.Conte, Carl de boor, *Elementary Numerical Analysis*, 3 Ed, McGraw-Hill, Newyork, 1980.
2. G. Howell, Derivative error bounds for Lagrange interpolation, *J. Approx, Th.*, 67 (1991) 164-173.
3. N.J.Higham, The Numerical Stability of Barycentric Lagrange Interpolation, *IMAJNA*, 24 (2004) 547-556.
4. N.J.Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelpia, 2002.
5. N.Mohankumar and S.M.Auerbach, On the use of higher order formulas for numerical derivatives in scientific computing, *Computer Physics Communications*, 161 (2004) 109–118.
6. A.Shadrin, Error bounds for Lagrange interpolation, *J. Approx. Th.*, 80 (1995) 25-49.