Journal of the Nigerian Association of Mathematical Physics
***Volume 31*, (July, *2015*), pp 347 – 362**

Comparative Analysis of Approximation Rules for Computing the Caputo Fractional Derivatives of Functions

Terhemen Aboiyar[1] and Sunday Simon Isah[2]

Department of Mathematics/Statistics/Computer Science, University of Agriculture, P.M.B. 2373, Makurdi, Nigeria

Abstract

In this paper, we propose two numerical methods for computing the Caputo fractional derivatives of functions by a weighted sum of function values at specified points. The first algorithm uses modified Trapezoidal rule in conjunction with a forward difference formula while the second algorithm uses the modified Trapezoidal rule in combination with a backward difference formula. Both the forward and backward difference formulas are of the second order. The error analysis for the approximation rules are presented. The approximation rules are implemented in MATLAB through some illustrative examples. Absolute errors are estimated and the orders of accuracy for the approximation rules are computed. The numerical experiments confirmed that all the approximation rules are accurate, efficient and readily implementable.

Keywords: Fractional integral; Trapezoidal Rule; Modified Trapezoidal Rule; Caputo Fractional Derivative.

1.0 Introduction

Fractional derivatives have wide applications in many areas especially in science and engineering. According to Atangana and Seer [1], "the standard mathematical models of integer-order derivatives, including non-linear models, do not work adequately in many cases. To this end, in recent years, fractional calculus has played a very important role in various fields such as mechanics, electricity, chemistry, biology, economics, notably control theory, signal and image processing".
Moreover, many real dynamic systems are better characterized using a non-integer order dynamic model based on fractional calculus or, differentiation or integration of non-integer order. Traditional Calculus is based on integer order differentiation and integration. Diethelm *et al*. [2] point out that "in recent years, it has turned out that many phenomena in engineering, physics, chemistry, and many other sciences can be described very successfully by models using mathematical tools from fractional calculus i.e., the theory of derivatives and integrals of fractional (non-integer) order". Some of the most prominent applications of fractional calculus are given in a book by Oldham and Spanier[3], in Caputo and Mainardi[4], the classical paper of Bagley and Torvik[5] as well as in the publications of Marks and Hall [6] and Olmstead and Handelsman[7]. More important results include the description of mechanical systems subject to damping [8], relaxation and reaction kinematics of polymers [9], so-called ultraslow processes [10], relaxation in filled polymer networks [11] as well as control theory [12].
Different models using fractional derivatives have been proposed and there has been significant interest in developing numerical schemes to find their approximated solution [13-19].
In this paper, we propose two approximation rules for computing the Caputo fractional derivatives of functions using forward and backward difference formulas which are a modification of the work of Odibat[20] which was based on centred difference approximations. The new algorithms are based on the use of the modified trapezoidal rule in conjunction with the difference approximations.

2.0 Methods

2.1 Fractional Derivatives

There are several definitions of fractional derivatives but we shall focus on the Caputo fractional derivative and Riemann-Liouville fractional derivative.

Corresponding author: Terhemen Aboiyar, E-mail: t_aboiyar@yahoo.co.uk, Tel.: +2347069121825 & 8068954264(S.S.I)

Definition 2.1: Caputo Fractional Derivative
Let m be the smallest integer that exceeds α, then Caputo fractional derivative of order $\alpha > 0$ is defined as:

$$D_*^{\alpha} f(x) = J^{m-\alpha}\left(f^{(m)}(x)\right) \tag{2.1}$$

$$= \begin{cases} \dfrac{1}{\Gamma(m-\alpha)}\left[\int_0^x \dfrac{f^{(m)}(t)}{(x-t)^{\alpha+1-m}}dt\right], & m-1<\alpha<m \\ \dfrac{d^m}{dx^m} f(t) & , \quad \alpha = m \end{cases} \tag{2.2}$$

where $m-1 < \alpha \le m \; and \; m \in N.$
The Caputo fractional derivative in (2.1) first computes an ordinary derivative of $f(x)$ followed by a fractional integral of the result to achieve the desired order of fractional derivative.

Definition 2.2: Riemann-Liouville Fractional Derivative
Let m be the smallest integer that exceeds α, then Riemann-Liouville fractional derivative of order $\alpha > 0$ is defined as:

$$D^{\alpha} f(\mathrm{x}) = \frac{d^m}{dx^m}\left(J^{m-\alpha} f(x)\right) \tag{2.3}$$

$$= \begin{cases} \dfrac{1}{\Gamma(m-\alpha)}\left(\dfrac{d}{dx}\right)^m \displaystyle\int_0^x \dfrac{f(t)}{(x-t)^{\alpha+1-m}}dt, & m-1<\alpha<m \\ \dfrac{d^m}{dx^m} f(t) & , \quad \alpha = m \end{cases} \tag{2.4}$$

where $m-1 < \alpha \le m \; and \; m \in N.$
The Riemann-Liouville fractional derivative in (2.3) first computes a fractional integral of $f(x)$ followed by an ordinary derivative of the result.

2.2 Numerical Differentiation

In this work, we shall employ only the central finite difference formulas for first and second derivatives, forward finite difference formulas for first and second derivatives as well as backward finite difference formulas for first and second derivatives respectively.

2.2.1 Central Difference Formulas

The central difference formulas for $f'(x_i)$ and $f''(x_i)$ respectively are given by:

$$f'(x_j) = \frac{f(x_j+h) - f(x_j-h)}{2h} + \boldsymbol{O}(h^2) \tag{2.5}$$

$$f''(x_j) = \frac{f(x_j+h) - 2f(x_j) + f(x_j-h)}{h^2} + \boldsymbol{O}(h^2) \tag{2.6}$$

More generally, the nth-order central finite difference formula for any integer *n* is given by:

$$f^{2n}(x_i) = \frac{1}{h^{2n}}\sum_{i=0}^{2n}(-1)^i \frac{(2n)!}{i!\,(2n-i)!} f(x_i + (n-i)h) + \boldsymbol{O}(h^2), \tag{2.7}$$

$$f^{2n+1}(x_i) = \frac{1}{h^{2n+1}}\sum_{i=0}^{2n+1}(-1)^i \frac{(2n+1)!}{i!\,(2n+1-i)!}\frac{1}{2}\left(f(x_i+(n-i)h) + f(x_i+(n+1-i)h)\right) + \boldsymbol{O}(h^2)\;[21,22]. \tag{2.8}$$

2.2.2 Forward Difference Formulas

The forward difference formulas for $f'(x_i)$ and $f''(x_i)$ respectively are given by:

$$f'(x_i) = \frac{-3f(x_i) + 4f(x_i+h) - f(x_i+2h)}{2h} + \boldsymbol{O}(h^2) \tag{2.9}$$

$$f''(x_i) = \frac{2f(x_i) - 5f(x_i+h) + 4f(x_i+2h) - f(x_i+3h)}{h^2} + \boldsymbol{O}(h^2) \tag{2.10}$$

In general, the nth-order forward finite difference formula for any integer *n* is given by Hildebrand [23]:

$$\Delta_h^n[f](x) = \frac{1}{(2h)^n}\sum_{i=0}^{n}(-1)^i \frac{(n)!}{i!\,(n-i)!} f(x+(n-i)h) + \boldsymbol{O}(h^2). \qquad (2.11)$$

2.2.3 Backward Difference Formulas

The backward difference formulas for $f^{'}(x_i)$ and $f^{''}(x_i)$ respectively are given by:

$$f^{'}(x_i) = \frac{f(x_i-2h) - 4f(x_i-h) + 3f(x_i)}{2h} + \boldsymbol{O}(h^2) \qquad (2.12)$$

$$f^{''}(x_i) = \frac{-f(x_i-3h) + 4f(x_i-2h) - 5f(x_i-h) + 2f(x_i)}{h^2} + \boldsymbol{O}(h^2) \qquad (2.13)$$

In a more general way, the nth-order backward finite difference formula for any integer n is given by:

$$\nabla_h^n[f](x) = \frac{1}{(2h)^n}\sum_{i=0}^{n}(-1)^i \frac{(n)!}{i!\,(n-i)!} f(x-ih) + \boldsymbol{O}(h^2) \quad [23] \qquad (2.14)$$

2.3 Numerical Integration

In this work, we shall consider the modified trapezoidal rule for numerical integration.

Theorem 2.1[24-25]

Suppose that $f \in C^2[0,T]$, $\tilde{f}_k$ is the piecewise interpolation for f with nodes chosen at the $t_i = ih$ with $h = \frac{T}{k}, i = 0,1,2,\cdots,k,$ then,

$$(i)\int_0^{t_k}(t_k-t)^{\alpha-1}\tilde{f}_k(t)dt = \sum_{i=0}^{k} a_{i,k}\cdot f(t_i), \qquad (2.15)$$

where,

$$a_{i,k} = \frac{h^\alpha}{\alpha(\alpha+1)}\begin{cases}(k-1)^{\alpha+1} - (k-1-\alpha)k^\alpha, & i = 0,\\ (k-i+1)^{\alpha+1} + (k-i-1)^{\alpha+1} - 2(k-i)^{\alpha+1}, & 1 \le i \le k-1,\\ 1, & i = k,\end{cases} \qquad (2.16)$$

$$(ii)\left|\int_0^{t_k}(t_k-t)^{\alpha-1}f(t)dt - \sum_{i=0}^{k} a_{i,k}\cdot f(t_i)\right| \le C_\alpha \left\|f^{''}\right\|_\infty t_k^\alpha h^2, \qquad (2.17)$$

for some constant C_α depending only on α.

2.4 Modified Trapezoidal Rule

We shall present a review of the modified trapezoidal rule that was introduced by Odibat[20]. The modified trapezoidal rule is used to approximate the fractional integral $J^\alpha f(x)$ by a weighted sum of function values at specified points.

Here, we give a generalisation of trapezoidal rule to approximate the fractional integral $J^\alpha f(x)$ of order $\alpha > 0$.The following theorem states the modified trapezoidal rule.

Theorem 2.2[20]

Suppose that the interval $[0,a]$ is sub-divided into k sub-intervals $[x_j, x_{j+1}]$ of equal width $h = \frac{a}{k}$ by using the nodes $x_j = jh$, $for\ j = 0,1,\cdots,k$.The modified trapezoidal rule:

$$T(f,h,\alpha) = [(k-1)^{\alpha+1} - (k-\alpha-1)k^\alpha]\frac{h^\alpha f(0)}{\Gamma(\alpha+2)} + \frac{h^\alpha f(a)}{\Gamma(\alpha+2)} + \sum_{j=1}^{k-1}[(k-j+1)^{\alpha+1} - 2(k-j)^{\alpha+1} + (k-j-1)^{\alpha+1}]\frac{h^\alpha f(x_j)}{\Gamma(\alpha+2)} \qquad (2.18)$$

is an approximation to fractional integral:

$$[J^\alpha f(x)](a) = T(f,h,\alpha) - E_T(f,h,\alpha), a > 0, \alpha > 0 \qquad (2.19)$$

Furthermore, if $f(x) \in C^2[0,a]$, there is a constant $C_\alpha^{'}$ depending only on α so that the error term $E_T(f,h,\alpha)$ has the form:
$|E_T(f,h,\alpha)| \le C_\alpha^{'}\left\|f^{''}\right\|_\infty a^\alpha h^2 = \boldsymbol{O}(h^2)$

2.5 Approximation of Caputo Fractional Derivative

We shall present a review of the Caputo fractional derivative rule that was introduced by Odibat[20]. The Caputo fractional derivative rule is used to approximate the fractional derivative $D_*^\alpha f(x)$ by a weighted sum of function ordinary derivatives values at specified points.
In this section, an algorithm to approximate Caputo fractional derivative of arbitrary order $\alpha > 0$ for a given function by a weighted sum of function and its ordinary derivatives values at specified points is derived. The algorithm is based on the approximation

$$\int_a^b f(x)dx \approx T(f,h) = \frac{h}{2}[f(a)+f(b)] + h\sum_{k=1}^{M-1} f(x_k).$$

The algorithm is stated in the next theorem.

Theorem 2.3[20]

Suppose that the interval $[0,a]$ is sub-divided into k sub-intervals $[x_j, x_{j+1}]$ of equal width, $h = \frac{a}{k}$ using the nodes $x_j = jh$, for $j = 0,1,2,\dots,k$, then the Caputo fractional derivative approximation rule for (2.1) – (2.2):

$$C(f,h,\alpha) = \frac{h^{m-\alpha}}{\Gamma(m-\alpha+2)}\left\{\begin{array}{l}[(k-1)^{m-\alpha+1} - (k-m+\alpha-1)k^{m-\alpha}]f^{(m)}(0) + f^{(m)}(a) \\ +\sum_{j=1}^{k-1}[(k-j+1)^{m-\alpha+1} - 2(k-j)^{m-\alpha+1} + (k-j-1)^{m-\alpha+1}]f^{(m)}(x_j)\end{array}\right. \quad (2.20)$$

is an approximation to the Caputo fractional derivative and

$$\left(D_*^\alpha f(x)\right)(a) = C(f,h,\alpha) - E_C(f,h,\alpha), a > 0, \quad (2.21)$$

for $m-1 < \alpha \le m$.

Furthermore, if $f(x) \in C^{m+2}[0,a]$, then there is some constant $C'_{m-\alpha}$ depending only on α so that the error term $E_C(f,h,\alpha)$ has the form

$$|E_C(f,h,\alpha)| \le C'_{m-\alpha}\|f^{m+2}\|_\infty a^{m-\alpha}h^2 = \boldsymbol{O}(h^2) \quad (2.22)$$

2.6 Approximation Rules

In this section, we present the step-by-step or the procedures that lead to the approximation rules that we have proposed for computing the Caputo fractional derivatives of functions. The approximation rules are based on the modified trapezoidal rule (2.18), approximation of Caputo fractional derivative(2.20)and the finite difference formulas[21-23]that are used to approximate the ordinary derivative $f^{(m)}(x_i)$.
Next, we propose the approximation rules for the numerical computations of the Caputo fractional derivatives of functions. Our approach is based on the finite difference formulas (2.5)-(2.14) and the Caputo fractional derivative approximation rule (2.20).

2.6.1 The Approximation Rule of Odibat[20] (AR1)

In this section, we present the approximation rule which was postulated by Odibat[20] for computing the fractional derivative of a function by a weighted sum of function values at specified points.
To approximate the fractional derivative $[D_*^\alpha f(x)](a), m-1 < \alpha \le m$, assume that the interval $[0,a]$ is sub-divided into k sub-intervals $[x_{j,}x_{j+1}]$ of equal width, $h = \frac{a}{k}$ by using the nodes $x_j = jh$, for $j = 0,1,2,\dots,k-1$. We first approximate the integral in $[D_*^\alpha f(x)](a)$ with the modified trapezoidal rule leading to (2.20). For small h, using the central finite difference formulas in (2.5), (2.6), (2.7) and (2.8), we can approximate the ordinary derivative $f^m(x_j)$.
Now, if we replace the term $f^{(m)}(x_j), m-1 < \alpha \le m$, on the right hand side of equation (2.20) with the appropriate formula from equations (2.5) - (2.8) then by cancelling the term h^m, we obtain the following approximation rule:

$$\begin{aligned}[D_*^\alpha f(x)](a) = \frac{h^{-\alpha}}{2\Gamma(3-\alpha)}\{[(k-1)^{2-\alpha} - (k+\alpha-2)k^{1-\alpha}]g_1(0) + g_1(a) \\ + \sum_{j=1}^{k-1}[(k-j+1)^{2-\alpha} - 2(k-j)^{2-\alpha} + (k-j-1)^{2-\alpha}]g_1(x_j)\} + E(f,h,\alpha), \\ 0 < \alpha \le 1 \end{aligned} \quad (2.23)$$

$$[D_*^\alpha f(x)](a) = \frac{h^{-\alpha}}{\Gamma(4-\alpha)}\{[(k-1)^{3-\alpha} - (k+\alpha-3)k^{2-\alpha}]g_2(0) + g_2(a) + \sum_{j=1}^{k-1}[(k-j+1)^{3-\alpha} - 2(k-j)^{3-\alpha} + (k-j-1)^{3-\alpha}]g_2(x_j)\} + E(f,h,\alpha),$$
$$1 < \alpha \le 2, \tag{2.24}$$

and in general, we have:

$$[D_*^\alpha f(x)](a) = \frac{h^{-\alpha}}{\Gamma(m+2-\alpha)}\{[(k-1)^{m-\alpha+1} - (k-m+\alpha-1)k^{m-\alpha}]g_m(0) + g_m(a) + \sum_{j=1}^{k-1}[(k-j+1)^{m-\alpha+1} - 2(k-j)^{m-\alpha+1} + (k-j-1)^{m-\alpha+1}]g_m(x_j)\} + E(f,h,\alpha),$$
$$m-1 < \alpha \le m, \tag{2.25}$$

where

$g_1(x) = f(x+h) - f(x-h),$

$g_2(x) = f(x+h) - 2f(x) + f(x-h),$

$$g_{2n}(x) = \sum_{j=0}^{2n}(-1)^j \frac{(2n)!}{j!\,(2n-j)!} f(x+(n-j)h),$$

$$g_{2n+1}(x) = \sum_{j=0}^{2n+1}(-1)^j \frac{(2n+1)!}{j!\,(2n+1-j)!}\frac{1}{2}(f(x+(n-j)h) + f(x+n+1-j)h)).$$

Furthermore, if $f(x) \in C^{m+2}[0,a]$, there is a constant C'_α depending only on α so that the error term $E(f,h,\alpha)$ has the form: $|E(f,h,\alpha)| \le C'_\alpha \left\|f^{(m+2)}\right\|_\infty a^{m-\alpha}h^2 = \boldsymbol{O}(h^2)$**.**

The above computational algorithms depend on trapezoidal rule for numerical integration of definite integrals as well as central difference formulas (2.5)-(2.8)for numerical differentiation of functions.

2.6.2 Approximation Rules based on Forward Difference and Backward Difference Formulas

In this sub-section, we propose two new computational algorithms for fractional derivatives of functions in Caputo sense as an extension of the work of Odibat[20].

I. Approximation Rule Two (AR2)

We now propose the first new approximation rule for the numerical computations of the fractional derivatives. Our approach is based on the finite difference formulas (2.9)-(2.11) and the Caputo fractional derivative approximation rule (2.20).

To approximate the fractional derivative$[D_*^\alpha p(x)](a), m_1 - 1 < \alpha \le m_1$, assume that the interval $[0,a]$ is sub-divided into d sub-intervals $[x_i, x_{i+1}]$ of equal width, $h = \frac{a}{d}$ by using the nodes $x_i = ih$, for $i = 0,1,2,\ldots,d$. It is known that, for small h, using the forward difference formulas (2.9)-(2.11), we can approximate the ordinary derivatives $p^{(m)}(x_i)$.

Now, if we replace the term $p^{(m)}(x_i), m_1 - 1 < \alpha \le m_1$, on the right hand side of equation (2.20) with the appropriate formula from equations (2.9) - (2.11) then by cancelling the term h^m, we obtain the following approximation rule:

$$[D_*^\alpha p(x)](a) = \frac{h^{-\alpha}}{2\Gamma(3-\alpha)}\{[(d-1)^{2-\alpha} - (d+\alpha-2)d^{1-\alpha}]p_1(0) + p_1(a) + \sum_{i=1}^{d-1}[(d-i+1)^{2-\alpha} - 2(d-i)^{2-\alpha} + (d-i-1)^{2-\alpha}]p_1(x_i)\} + E_1(p,h,\alpha),$$
$$0 < \alpha \le 1 \tag{2.26}$$

$$[D_*^\alpha p(x)](a) = \frac{h^{-\alpha}}{\Gamma(4-\alpha)}\{[(d-1)^{3-\alpha} - (d+\alpha-3)d^{2-\alpha}]p_2(0) + p_2(a) + \sum_{i=1}^{d-1}[(d-i+1)^{3-\alpha} - 2(d-i)^{3-\alpha} + (d-i-1)^{3-\alpha}]p_2(x_i)\} + E_1(p,h,\alpha),$$
$$1 < \alpha \le 2, \tag{2.27}$$

and in general, we have:

$$[D_*^{\alpha}p(x)](a)=\frac{h^{-\alpha}}{\Gamma(m_1+2-\alpha)}\{[(d-1)^{m_1-\alpha+1}-(d-m_1+\alpha-1)d^{m_1-\alpha}]p_{m_1}(0)+p_{m_1}(a)$$
$$+\sum_{i=1}^{d-1}[(d-i+1)^{m_1-\alpha+1}-2(d-i)^{m_1-\alpha+1}+(d-i-1)^{m_1-\alpha+1}]p_{m_1}(x_i)\}+E_1(p,h,\alpha),$$
$$m_1-1<\alpha\le m_1, \tag{2.28}$$

where

$p_1(x)=-3p(x)+4p(x+h)-p(x+2h),$

$p_2(x)=2p(x)-5p(x+h)+4p(x+2h)-p(x+3h),$

$$p_n(x)=\sum_{i=0}^{n}(-1)^i\frac{(n)!}{i!\,(n-i)!}p(x+(n-i)h).$$

The above approximation rule solely depends on trapezoidal rule (2.20) for numerical integration of definite integrals as well as forward difference formulas (2.9)-(2.11) for numerical differentiation of functions.

II. Approximation Rule Three (AR3)

In this sub-section, like before, we equally propose the second new approximation rule for the numerical computations of the fractional derivative. Our approach here is based on the backward finite difference formulas (2.12)-(2.14) and the Caputo fractional derivative approximation rule (2.20).

To approximate the fractional derivative $[D_*^{\alpha}q(x)](a), m_2-1<\alpha\le m_2$, assume that the interval $[0,a]$ is sub-divided into r sub-intervals $[x_i,x_{i+1}]$ of equal width, $h=\frac{a}{r}$ by using the nodes $x_i=ih$, for $i=0,1,2,\dots,r$. It is known that, for small h, using the backward difference formulas (2.12)-(2.14), we can approximate the ordinary derivatives $q^{(m)}(x_i)$.

Now, if we replace the term $q^{(m)}(x_i), m_2-1<\alpha\le m_2$, on the right hand side of equation (2.20) with the appropriate formula from equations (2.12)-(2.14) then by cancelling the term h^m, we obtain the following approximation rules:

$$[D_*^{\alpha}q(x)](a)=\frac{h^{-\alpha}}{2\Gamma(3-\alpha)}\{[(r-1)^{2-\alpha}-(r+\alpha-2)r^{1-\alpha}]q_1(0)+q_1(a)$$
$$+\sum_{i=1}^{r-1}[(r-i+1)^{2-\alpha}-2(r-i)^{2-\alpha}+(r-i-1)^{2-\alpha}]q_1(x_i)\}+E_2(q,h,\alpha),$$
$$0<\alpha\le 1 \tag{2.29}$$

$$[D_*^{\alpha}q(x)](a)=\frac{h^{-\alpha}}{\Gamma(4-\alpha)}\{[(r-1)^{3-\alpha}-(r+\alpha-3)r^{2-\alpha}]q_2(0)+q_2(a)$$
$$+\sum_{i=1}^{r-1}[(r-i+1)^{3-\alpha}-2(r-i)^{3-\alpha}+(r-i-1)^{3-\alpha}]q_2(x_i)\}+E_2(q,h,\alpha),$$
$$1<\alpha\le 2, \tag{2.30}$$

and in general, we have:

$$[D_*^{\alpha}q(x)](a)=\frac{h^{-\alpha}}{\Gamma(m_2+2-\alpha)}\{[(r-1)^{m_2-\alpha+1}-(r-m_2+\alpha-1)r^{m_2-\alpha}]q_{m_2}(0)+q_{m_2}(a)$$
$$+\sum_{i=1}^{r-1}[(r-i+1)^{m_2-\alpha+1}-2(r-i)^{m_2-\alpha+1}+(r-i-1)^{m_2-\alpha+1}]q_m(x_i)\}+E_2(q,h,\alpha),$$
$$m_2-1<\alpha\le m_2, \tag{2.31}$$

where

$q_1(x)=q(x-2h)-4q(x-h)+3q(x),$

$q_2(x)=-q(x-3h)+4q(x-2h)-5q(x-h)+2q(x),$

$$q_n(x)=\sum_{i=0}^{n}(-1)^i\frac{(n)!}{i!\,(n-i)!}f(x-ih).$$

The above approximation rule three depends on trapezoidal rule (2.20) for numerical integration of definite integrals as well as backward difference formulas (2.12)-(2.14) for numerical differentiation of functions.

The remaining part of this Chapter will be devoted to the error analysis for the three numerical schemes (the existing method and the two newly proposed methods).

3.0 Results and Discussion

3.1 Error Analysis for the Numerical Schemes

This section will be devoted to the error analysis for the three approximation rules (the existing method and the two newly proposed methods).

3.1.1. Error Analysis for the First Numerical Scheme

The error estimate for the first approximation is given in Odibat[20].

3.1.2. Error Analysis for the Second Numerical Scheme

In the same vein, the error estimate for the second approximation is given in the theorem below. The result follows from the work of Odibat[20]:

Theorem 3.1

Supposing that $f(x) \in C^{m_1+4}[-\epsilon, a+\epsilon]$, where $\epsilon > dh$ if $2(d-1) < \alpha \leq 2d$, $d = 1, 2, \ldots$, then the truncation error $E_1(p, h, \alpha)$ has the form

$$|E_1(p,h,\alpha)| = \boldsymbol{O}(h^2), \tag{3.1}$$

where $m_1 - 1 < \alpha \leq m_1$.

Proof

Starting with the second-degree Taylor expansions about x, for $p''(x_i + dh)$:

$$p''(x_i + dh) = \sum_{i=0}^{\infty} \frac{(dh)^i}{i!} p^{(i+2)}(x_i), \tag{3.2}$$

From the expansions (3.2), since $p(x) \in C^{m_1+4}[-\epsilon, a+\epsilon]$**,** we can obtain the following formulas:

$$-3p''(x_i) + 4p''(x_i + h) - p''(x_i + 2h) = 2hp^{(3)}(x_i) + \boldsymbol{O}(h^3), \tag{3.3}$$

$$2p''(x_i) - 5p''(x_i + h) + 4p''(x_i + 2h) - p''(x_i + 3h) = h^2 p^4(x_i) + \boldsymbol{O}(h^4), \tag{3.4}$$

$$\sum_{i=0}^{n} (-1)^i \frac{(n)!}{i!\,(n-i)!} p''(x_i + (n-i)h) = h^n p^{(n+2)}(x_i) + \boldsymbol{O}(h^{n+2}) \tag{3.5}$$

Comparing (2.28) with (2.20), we can observe that

$$[D_*^{\alpha} p](a) = C(\theta, h, \alpha) + E_1(p, h, \alpha), \qquad \text{where } \theta^{(m_1)} = h^{-m_1} p_{m_1}. \tag{3.6}$$

Therefore, using (2.22), we obtain

$$\begin{aligned} |E_1(p,h,\alpha)| &\leq C'_\alpha \left\| \theta^{(m_1+2)} \right\|_\infty a^{m_1-\alpha} h^2, \\ &= C'_\alpha \left\| p''_{m_1} \right\|_\infty a^{m_1-\alpha} h^{2-m_1}, \end{aligned} \tag{3.7}$$

and using (3.3)-(3.5), we get

$$\begin{aligned} |E_1(p,h,\alpha)| &\leq C'_\alpha \left\| p^{(m_1+2)} + \boldsymbol{O}(h^2) \right\|_\infty a^{m_1-\alpha} h^2, \\ &= \boldsymbol{O}(h^2). \end{aligned} \tag{3.8}$$

3.3.3. Error Analysis for the Third Numerical Scheme

Here, the error estimate for the third approximation is given in the theorem below:

Theorem 3.2

Supposing that $f(x) \in C^{m_2+4}[-\epsilon, a+\epsilon]$, where $\epsilon > rh$ if $2(r-1) < \alpha \leq 2r$, $r = 1, 2, \ldots$, then the truncation error $E_2(q, h, \alpha)$ has the form

$$|E_2(q,h,\alpha)| = \boldsymbol{O}(h^2), \tag{3.9}$$

where $m_2 - 1 < \alpha \leq m_2$.

Proof

Starting with the second-degree Taylor expansions about x, for $q''(x_i - rh)$:

$$q''(x_i - rh) = \sum_{i=0}^{\infty} (-1) \frac{(rh)^i}{i!} q^{(i+2)}(x_i), \tag{3.10}$$

From the expansions (3.10), since $q(x) \in C^{m_2+4}[-\epsilon, a+\epsilon]$**,** we can obtain the following formulas:

$$q''(x_i - 2h) - 4q''(x_i - h) + 3q''(x_i) = 2hq^{(3)}(x_i) + \boldsymbol{O}(h^3), \tag{3.11}$$

$$-q''(x_i - 3h) + 4p''(x_i - 2h) - 5q''(x_i - h) + 2q''(x_i) = h^2 q^4(x_i) + \boldsymbol{O}(h^4), \tag{3.12}$$

$$\sum_{i=0}^{n} (-1)^i \frac{(n)!}{i!\,(n-i)!} q''(x_i - ih) = h^n q^{(n+2)}(x_i) + \boldsymbol{O}(h^{n+2}), \tag{3.13}$$

Comparing (2.31) with (2.20), we can observe that

$[D_*^\alpha q](a) = C(\delta, h, \alpha) + E_2(q, h, \alpha)$, where $\delta^{(m_2)} = h^{-m_2} q_{m_2}$. (3.14)

Therefore, using (2.22), we obtain

$$|E_2(p, h, \alpha)| \leq C'_\alpha \left\| \delta^{(m_2+2)} \right\|_\infty a^{m_2-\alpha} h^2,$$

$$= C'_\alpha \left\| q''_{m_1} \right\|_\infty a^{m_2-\alpha} h^{2-m_2}, \tag{3.15}$$

and using (3.11)-(3.13), we get

$$|E_2(q, h, \alpha)| \leq C'_\alpha \left\| q^{(m_2+2)} + \boldsymbol{O}(h^2) \right\|_\infty a^{m_2-\alpha} h^2,$$
$$= \boldsymbol{O}(h^2). \tag{3.16}$$

3.2 Numerical Experiments

In this section, we consider some numerical examples to illustrate the three approximation rules as numerical tools. These examples are somewhat artificial in the sense that the exact values of fractional derivatives are known in advance. Nevertheless, such an approach is needed to examine the accuracy and the efficiency of the approximation rules for the purpose of applying them for problems where the exact value of the derivative is not known.

Besides, for the purpose of this work, we use the small Greek letter kappa (κ) for order of accuracy (κ). The Order of Accuracy (κ) is mathematically defined as:

$$\text{Order of Accuracy } (\kappa) = \frac{\log\left(\frac{E(h)}{E\left(\frac{h}{2}\right)}\right)}{\log(2)}$$

where

$E =$ absolute error for the approximation rule one;

$h =$ step size.

3.2.1 Example1

Consider the function $f(x) = \sin(x)$. We approximate the fractional derivative $[D_*^\alpha \sin(x)]$ for $\alpha = 0.1$, $\alpha = 0.5$, $\alpha = 1$ and $\alpha = 1.5$ and $[0,1]$. We use $h = \frac{a}{k} = \frac{a}{d} = \frac{a}{r}$; where $k = d = r = 10,\ 20, 40,\ 80,\ 160,$ and 320; $x_j = jh$ and $x_i = ih$.

Using the definition of Caputo fractional derivative (3.1) and the formulas (3.8) and (3.9), the true value of the Caputo fractional derivative $D_*^\alpha \sin(x)$ is given by

$$(i) \quad D_*^\alpha \sin(x) = x^{1-\alpha} \sum_{k=0}^{\infty} \frac{(-1)^k (x)^{2k}}{\Gamma(2k - \alpha + 2)}, \qquad for\ 0 < \alpha \leq 1 \tag{3.17}$$

and

$$(ii) \quad D_*^\alpha \sin(x) = x^{2-\alpha} \sum_{k=0}^{\infty} \frac{(-1)^k (x)^{2k+1}}{\Gamma(2k - \alpha + 4)}, \qquad for\ 1 < \alpha \leq 2 \tag{3.18}$$

Note that the true value of $D_*^\alpha \sin(x)$ is calculated when $x = 1$.

Table 3.1: Approximate Values of $[D_*^{0.1}\sin(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{0.1}\sin(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	0.86068646	0.85853763	0.86270193	0.86295155	2.1488×10^{-3}	2.0155×10^{-3}	2.2651×10^{-3}	-	-	-
20	0.050000	0.86068646	0.86014880	0.86120813	0.86123939	5.3766×10^{-4}	5.2167×10^{-4}	5.5293×10^{-4}	1.9988	1.9499	2.0344
40	0.025000	0.86068646	0.86055200	0.86081896	0.86082287	1.3445×10^{-4}	1.3250×10^{-4}	1.3641×10^{-4}	1.9996	1.9771	2.0191
80	0.012500	0.86068646	0.86065284	0.86071983	0.86072032	3.3620×10^{-5}	3.3380×10^{-5}	3.3860×10^{-5}	1.9997	1.9889	2.0103
160	0.006250	0.86068646	0.86067805	0.86069483	0.86069489	8.4000×10^{-6}	8.3700×10^{-6}	8.4400×10^{-6}	2.0009	1.9957	2.0043
320	0.003125	0.86068646	0.86068436	0.86068856	0.86068856	2.1000×10^{-6}	2.1000×10^{-6}	2.1100×10^{-6}	2.0000	1.9948	2.0000

Table 3.2: Approximate Values of $[D_*^{0.5}\sin(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{0.5}\sin(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	0.84605679	0.84397472	0.84802406	0.84835809	2.0821×10^{-3}	1.9673×10^{-3}	2.3013×10^{-3}	-	-	-
20	0.050000	0.84605679	0.84553377	0.84656955	0.84661138	5.2302×10^{-4}	5.1276×10^{-4}	5.5459×10^{-4}	1.9931	1.9398	2.0530
40	0.025000	0.84605679	0.84592561	0.84618733	0.84619256	1.3118×10^{-4}	1.3054×10^{-4}	1.3578×10^{-4}	1.9953	1.9738	2.0302
80	0.012500	0.84605679	0.84602392	0.84608969	0.84609034	3.2870×10^{-5}	3.2900×10^{-5}	3.3550×10^{-5}	1.9967	1.9883	2.0169
160	0.006250	0.84605679	0.84604856	0.84606504	0.84606512	8.2300×10^{-6}	8.2500×10^{-6}	8.3300×10^{-6}	1.9978	1.9956	2.0099
320	0.003125	0.84605679	0.84605473	0.84605885	0.84605886	2.0600×10^{-6}	2.0700×10^{-6}	2.0800×10^{-6}	1.9982	1.9948	2.0017

Table 3.3: Approximate Values of $[D_*^1 \sin(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^1 \sin(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	0.54030231	0.53940225	0.54188700	0.54230703	9.0005×10^{-4}	1.5847×10^{-3}	2.00473×10^{-3}	-	-	-
20	0.050000	0.54030231	0.54007721	0.54072588	0.54077845	2.2510×10^{-4}	4.2357×10^{-4}	47614×10^{-4}	1.9994	1.9035	2.0740
40	0.025000	0.54030231	0.54024603	0.54041156	0.54041813	5.6280×10^{-5}	1.0925×10^{-4}	1.1583×10^{-4}	1.9999	1.9550	2.0394
80	0.012500	0.54030231	0.54028824	0.54033003	0.54033086	1.4070×10^{-5}	2.7730×10^{-5}	2.8550×10^{-5}	2.0000	1.9781	2.0204
160	0.006250	0.54030231	0.54029879	0.54030929	0.54030939	3.5200×10^{-5}	6.9800×10^{-6}	7.0900×10^{-6}	1.9990	1.9901	2.0096
320	0.003125	0.54030231	0.54030143	0.54030406	0.54030407	8.8000×10^{-4}	1.7500×10^{-6}	1.7700×10^{-6}	2.0000	1.9959	2.0020

Table 3.4: Approximate Values of $[D_*^{1.5} \sin(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{1.5} \sin(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	-0.66968426	-0.66862079	-0.67611062	-0.67442548	1.0635×10^{-3}	6.4264×10^{-3}	4.7412×10^{-3}	-	-	-
20	0.050000	-0.66968426	-0.66941438	-0.67119112	-0.67097982	2.6988×10^{-4}	1.5069×10^{-3}	1.2956×10^{-3}	1.9784	2.0925	1.8717
40	0.025000	-0.66968426	-0.66961611	-0.67004768	-0.67002125	6.8150×10^{-5}	3.6342×10^{-4}	3.3699×10^{-4}	1.9855	2.0518	1.9428
80	0.012500	-0.66968426	-0.6696671	-0.66977338	-0.66977008	1.7160×10^{-5}	8.9120×10^{-5}	8.5820×10^{-5}	1.9897	2.0278	1.9733
160	0.006250	-0.66968426	-0.66967995	-0.66970631	-0.6697059	4.3100×10^{-6}	2.2060×10^{-5}	2.1640×10^{-5}	1.9933	2.0143	1.9876
320	0.003125	-0.66968426	-0.66968318	-0.66968974	-0.66968969	1.0800×10^{-4}	5.4800×10^{-6}	5.4300×10^{-6}	1.9967	2.0092	1.9947

Tables 3.1 to 3.4 show the exact values and approximate values of the fractional derivative $[D_*^{\alpha}\sin(x)](1)$, the errors for $\alpha = 0.1$, $\alpha = 0.5$, $\alpha = 1$ and $\alpha = 1.5$ and the order of accuracy for the three approximation rules (i.e., the approximation rule based on central difference formulas, the approximation rule based on forward difference formulas and the approximation rule based on backward difference formulas). From the numerical results, we observed the following salient points:

- As the step size gets smaller, all the methods converge. However, for $\alpha = 0.1$, AR1 is best, for $\alpha = 0.5$,AR2 is best, $\alpha = 1$, AR1 is best, for $\alpha = 1.5$, AR1is best.
- We observed from the Tables 3.1-3.4 that the computed order of accuracy (κ) is approximately two, which is in agreement with the error analysis that is of order two ($\boldsymbol{O}(h^2)$) that is, when the step size h is reduced by a factor of $\frac{1}{2}$, the successive absolute errors are diminished by $\frac{1}{4}$.

3.2.2 Example 2

Consider the function $f(x) = \cos(x)$. We approximate the fractional derivative $[D_*^{\alpha}\cos(x)]$ for $\alpha = 0.1$, $\alpha = 0.5$, $\alpha = 1$ and $\alpha = 1.5$ and $[0,1]$.We use $h = \frac{a}{k} = \frac{a}{d} = \frac{a}{r}$; where $k = d = r = 10,\ 20, 40,\ 80,\ 160,$ and 320; $x_j = jh$ and $x_i = ih$.

Using the definition of Caputo fractional derivative (1.7) and the formula (3.8) and (3.9), the true value of the fractional derivative $D_*^{\alpha}\cos(x)$ is given by

$$(i) \quad D_*^{\alpha}\cos(x) = x^{1-\alpha}\sum_{k=0}^{\infty}\frac{(-1)^k(x)^{2k+1}}{\Gamma(2k-\alpha+3)}, \qquad for\ 0 < \alpha \leq 1 \tag{3.19}$$

$$(ii) \quad D_*^{\alpha}\cos(x) = x^{2-\alpha}\sum_{k=0}^{\infty}\frac{(-1)^k(x)^{2k}}{\Gamma(2k-\alpha+3)}, \qquad for\ 1 < \alpha \leq 2 \tag{3.20}$$

Note that the true value of $D_*^{\alpha}\cos(x)$ is calculated when $x = 1$.

Table 3.5: Approximate Values of $[D_*^{0.1}\cos(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{0.1}\cos(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	-0.50049690	-0.49924881	-0.50195761	-0.50152834	1.2481×10^{-3}	1.4607×10^{-3}	1.0314×10^{-3}	-	-	-
20	0.050000	-0.50049690	-0.50018443	-0.50083641	-0.50078265	3.1247×10^{-4}	3.3951×10^{-4}	2.8575×10^{-4}	1.9979	2.1051	1.8518
40	0.025000	-0.50049690	-0.50041874	-0.50057848	-0.50057175	7.8160×10^{-5}	8.1570×10^{-5}	7.4850×10^{-5}	1.9992	2.0573	1.9327
80	0.012500	-0.50049690	-0.50047736	-0.50051688	-0.50051604	1.9550×10^{-5}	1.9970×10^{-5}	1.9130×10^{-5}	1.9993	2.0302	1.9682
160	0.006250	-0.50049690	-0.50049202	-0.50050184	-0.50050174	4.8900×10^{-6}	4.9400×10^{-6}	4.8400×10^{-6}	1.9993	2.0153	1.9828
320	0.003125	-0.50049690	-0.50049568	-0.50049813	-0.50049812	1.2200×10^{-6}	1.2300×10^{-6}	1.2200×10^{-6}	2.0030	2.0059	1.9881

Table 3.6: Approximate Values of $[D_*^{0.5}\cos(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{0.5}\cos(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	-0.66968426	-0.66806351	-0.67161204	-0.67119005	1.6208×10^{-3}	1.9278×10^{-3}	1.5058×10^{-3}	-	-	-
20	0.050000	-0.66968426	-0.66927491	-0.67013775	-0.67008490	4.0935×10^{-4}	4.5349×10^{-4}	4.0064×10^{-4}	1.9853	2.0878	1.9101
40	0.025000	-0.66968426	-0.66958123	-0.66979377	-0.66978716	1.0303×10^{-4}	1.0951×10^{-4}	1.0290×10^{-4}	1.9903	2.0500	1.9611
80	0.012500	-0.66968426	-0.66965838	-0.66971111	-0.66971029	2.5880×10^{-5}	2.6850×10^{-5}	2.6030×10^{-5}	1.9932	2.0281	1.9830
160	0.006250	-0.66968426	-0.66967777	-0.66969090	-0.66969080	6.4900×10^{-6}	6.6400×10^{-6}	6.5400×10^{-6}	1.9955	2.0157	1.9928
320	0.003125	-0.66968426	-0.66968263	-0.66968591	-0.66968590	1.6300×10^{-6}	1.6500×10^{-6}	1.6400×10^{-6}	1.9933	2.0087	1.9956

Table 3.7: Approximate Values of $[D_*^1 \cos(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^1 \cos(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	-0.84147098	-0.84006923	-0.84440093	-0.84413123	1.4018×10^{-3}	2.9300×10^{-3}	2.6603×10^{-3}	-	-	-
20	0.050000	-0.84147098	-0.84112042	-0.84218847	-0.84215472	3.5057×10^{-4}	7.1749×10^{-4}	6.8373×10^{-4}	1.9995	2.0298	1.9601
40	0.025000	-0.84147098	-0.84138333	-0.84164836	-0.84164414	8.7650×10^{-5}	1.7738×10^{-4}	1.7316×10^{-4}	1.9999	2.0161	1.9813
80	0.012500	-0.84147098	-0.84144907	-0.84151507	-0.84151455	2.1910×10^{-5}	4.4090×10^{-5}	4.3560×10^{-5}	2.0002	2.0083	1.9910
160	0.006250	-0.84147098	-0.84146551	-0.84148197	-0.84148191	5.4800×10^{-6}	1.0990×10^{-5}	1.0920×10^{-5}	1.9993	2.0043	1.9960
320	0.003125	-0.84147098	-0.84146962	-0.84147373	-0.84147372	1.3700×10^{-6}	2.7400×10^{-6}	2.7400×10^{-6}	2.0000	2.0039	1.9947

Table 3.8: Approximate Values of $[D_*^{1.5} \cos(x)](1)$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

k	h	exact value	$[D_*^{1.5} \cos(x)](1)$			absolute errors			order of accuracy (κ)		
			AR1	AR2	AR3	AR1	AR2	AR3	AR1	AR2	AR3
10	0.100000	-0.84605679	-0.84467874	-0.85240937	-0.85374327	1.3781×10^{-3}	6.3526×10^{-3}	7.6865×10^{-3}	-	-	-
20	0.050000	-0.84605679	-0.84570997	-0.84773753	-0.84790478	3.4682×10^{-4}	1.6808×10^{-3}	1.8480×10^{-3}	1.9904	1.9182	2.0564
40	0.025000	-0.84605679	-0.84596967	-0.84648775	-0.84650867	8.7120×10^{-5}	4.3096×10^{-4}	4.5188×10^{-4}	1.9931	1.9635	2.032
80	0.012500	-0.84605679	-0.84603493	-0.84616581	-0.84616842	2.1850×10^{-5}	1.0902×10^{-4}	1.1164×10^{-4}	1.9954	1.9830	2.0171
160	0.006250	-0.84605679	-0.84605131	-0.8460842	-0.84608452	5.4800×10^{-6}	2.7410×10^{-5}	2.7740×10^{-5}	1.9954	1.9918	2.0088
320	0.003125	-0.84605679	-0.84605542	-0.84606366	-0.84606370	1.3700×10^{-6}	6.8700×10^{-6}	6.9100×10^{-6}	2.0000	1.9963	2.0052

Tables 3.5 to 3.8 present the true values and approximate values of the fractional derivative $[D_*^{\alpha}\cos(x)](1)$, the errors for $\alpha = 0.1$, $\alpha = 0.5$, $\alpha = 1$ and $\alpha = 1.5$ and the order of accuracy for the three approximation rules (i.e., the approximation rule based on central difference formulas, the approximation rule based on forward difference formulas and the approximation rule based on backward difference formulas). From the numerical results, we observed the following conspicuous points:

- Similar to example 1, as the step size gets smaller, all the methods converge. However, for $\alpha = 0.1$, AR3 is best, for $\alpha = 0.5$, AR3 is best, $\alpha = 1$, AR1 is best, for $\alpha = 1.5$, AR1is best.
- We observed from the Tables 3.5-3.8 that the computed order of accuracy (κ) is approximately two, which is in agreement with the error analysis that is of order two ($\boldsymbol{O}(h^2)$) that is, when the step size h is reduced by a factor of $\frac{1}{2}$, the successive absolute errors are diminished by $\frac{1}{4}$.

3.2 Computing Time for the Three Approximation Rules

The Table below presents the computing time in seconds for the three approximation rules (i.e., the approximation rule based on central difference formulas, the approximation rule based on forward difference formulas and the approximation rule based on backward difference formulas).

Table 3.9: Computing Time at $k = 10$ and $h = 0.1$ for Approximation Rule One (AR1), Approximation Rule Two (AR2) and Approximation Rule Three (AR3).

$\boldsymbol{\alpha}$	computing time (seconds)		
	AR1	AR2	AR3
0.1	0.019388	0.013725	0.027231
0.3	0.007875	0.007729	0.013079
0.5	0.007778	0.017480	0.017551
1.0	0.007761	0.008219	0.009027
1.3	0.008005	0.021919	0.018808
1.5	0.007832	0.008105	0.014307
1.7	0.007571	0.007815	0.015232
2.0	0.008072	0.007791	0.016779

From Table3.9, we observed that the computational time for the three algorithmsis about the same.

4.0 Conclusion

In this paper, we implemented the computational algorithm for computing the fractional derivatives of functions which was based on the trapezoidal rule in conjunction with the central difference formula for derivatives of functions that was proposed by Odibat [20]. We further developed and implemented two other computational algorithms for computing the fractional derivatives of functions which were based on trapezoidal rule in conjunction with:

i. the forward difference formulas for derivatives of functions; and
ii. the backward difference formulas for derivatives of functions respectively.

With the help of the two functions chosen as examples, we demonstrated the functionality and the efficiency of the three approximation rules. Tables 3.1 – 3.8 present to us the nature and the behavior of each of the three approximation rules. The results for the two algorithms we have proposed compare favourably with the existing results and in some instances provide more accurate results (see Table 3.4). Hence, we recommend that any of the three approximation rules presented in this paper can be employed for the computation of fractional derivatives of functions.

5.0 References

[1] Atangana, A. and Seer, A. (2013). A Note on Fractional Order Derivatives and Table of Fractional Derivatives of some special Functions, *Abstract and Applied Analysis*, Vol. 8,Pp. 2013.

[2] Diethelm, K., Ford, N. J. and Luchko, Y. (2005). Algorithms for the Fractional Calculus: A Selection of Numerical Methods, *Computational Methods for Applied Mechanical Engineering*, Vol.194, Pp.743-773.

[3] Oldham, K. B. and Spanier, J. (1974). *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, Inc, London.

[4] Caputo, M. and Mainardi, F. (1971). Linear Models of Dissipation in An elastic Solids. *Rivistadel Nuovo Cimento*, Vol.1, Pp. 161-198.

[5] Bagley, R. L. and Torvik, P. J. (1983). A Theoretical Basis for the Application of Fractional Calculus to Viscoelasticity, *Journal of Rheology* Vol. 27,Pp. 201-210.

[6] Marks, R. J. and Hall, M. W. (1981). Differintegral Interpolation from a Bandlimited Signal Samples, *IEEE Transmission Acoustic Speech Signal Process*, Vol. 29, Pp. 872-877.

[7] Olmstead, W. E. and Handelsman, R. A. (1976). Diffusion in a Semi-infinite Region with Nonlinear Surface Dissipation, *SIAMRev.*, Academic Press, Inc, Vol. 18, Pp. 275-291.

[8] Gaul, L., Klein, P. and Kempfle, S. (1991). Damping Description Involving Fractional Operators, *Mechanical System Signal Process*, Vol.5, Pp. 81-88.

[9] Glöckle, W. G. and Nonnenmacher, T. F. (1995). A Fractional Calculus Approach to Self-similar Protein Dynamics, *Biophysics Journal*, Vol.68, Pp. 46-53.

[10] Gorenflo, R. and Rutman, R. (1995). On Ultraslow and Intermediate Processes, *Science Culture Technology, Singapore*, Vol.1, Pp. 61-81.

[11] Metzler, R., Schick, W., Kilian, H. G. and Nonnenmacher, T. F. (1995). Relaxation in Filled Polymers: A Fractional Calculus Approach, *Journal of Chemical Physics*, Vol.103, Pp. 7180-7186.

[12] Podlubny, I. (1999). *Fractional Differential Equations*, Academic Press,San Diego.

[13] Liu, D.Y., Laleg-Kirati, T.M., Gibaru, O. and Perruquetti, W. (2013). Fractional Order Numerical Differentiation with B-Spline Functions,*The International Conference on Fractional Signals and Systems, Vol.2013*, Pp. 340-345.

[14] Brzezinski, D.R. and Ostalczyk, P. (2014). High-Accuracy Numerical Integration Methods for Fractional Order Derivatives and Integrals Computations, *Technical Sciences*, Vol. 62,No. 4, Pp. 723-733.

[15] Hasegawa, T. and Sugiura, H. (2009). Uniform Approximation to Fractional Derivatives of Functions of Algebraic Singularity,*Journal of Computational and Applied Mathematics*, Vol.228, Pp. 247-253.

[16] Maleknejad, K., Sahlan, M.N. and Ostadi, A. (2013). Numerical Solution of Fractional Integro-differential Equation by Using Cubic B-spline Wavelets, *Proceedings of the World Congress on Engineering*, Vol.1, Pp. 300-306.

[17] Machado, J.A.T. and Galhano, A.M.S. (2008). Numerical Calculation of Fractional Derivatives of Non-smooth Data,*ENOC*, Vol.1, Pp. 400-405.

[18] Sousa, E. (2010). How to Approximate the Fractional Derivative of Order $1 < \alpha \leq 2$. *Proceedings of FDA'10. The 4th IFAC Workshop Fractional Differentiation and Its Applications, Vol.*10, No. 019, 1-6.

[19] Li, M., Xiong, X.T. and Wang, Y.J. (2011). A Numerical Evaluation and Regularization of Caputo Fractional Derivatives,*Journal of Physics*, Vol.290, Pp.1-6.

[20] Odibat, Z. M. (2009). Computational Algorithms for Computing the Fractional Derivatives of Functions,*Mathematics and Computers in Simulation*, Vol.79, Pp. 2013-2020.

[21] Mathews, J. and Fink, K. (1990). *Numerical Methods Using MATLAB*, Prentice-Hall, USA.

[22] Gerald, C. and Wheatley, K. (2004). *Applied Numerical Analysis*, Addison-Wesley, USA.

[23] Hildebrand, F.B. (1968). Finite Difference Equations and Simulations, Prentice-Hall, New Jersey.

[24] Diethelm, K., Ford, N. J. and Freed, A. D. (2004). Detailed Error Analysis for a Fractional Adams Method, *Numerical Algorithms,* Vol.36, Pp.31-52.

[25] Odibat, Z. (2006). Approximations of Fractional Integrals and Caputo Fractional Derivatives, Applied *Mathematics and Computation,* Vol. 178, Pp. 527-533