BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 4, 2014
DOI: 10.2478/bpasts-2014-0078

High-accuracy numerical integration methods for fractional order derivatives and integrals computations

D.W. BRZEZIŃSKI* and P. OSTALCZYK

Faculty of Electrical, Electronic, Computer and Control Engineering, Institute Of Applied Computer Science,
Lodz University of Technology, 18/22 Stefanowskiego St., 90-537 Łódź, Poland

Abstract. In this paper the authors present highly accurate and remarkably efficient computational methods for fractional order derivatives and integrals applying Riemann-Liouville and Caputo formulae: the Gauss-Jacobi Quadrature with adopted weight function, the Double Exponential Formula, applying two arbitrary precision and exact rounding mathematical libraries (GNU GMP and GNU MPFR). Example fractional order derivatives and integrals of some elementary functions are calculated. Resulting accuracy is compared with accuracy achieved by applying widely known methods of numerical integration. Finally, presented methods are applied to solve Abel's Integral equation (in Appendix).

Key words: accuracy of numerical calculations, fractional order derivatives and integrals, double exponential formula, gauss-jacobi quadrature with adopted weight function, arbitrary precision, numerical integration, abel's integral equation.

1. Introduction

There are several formulae [1–3] which can be used to compute integrals and derivatives of non-integer (fractional) orders. They include the Riemann-Liouville/Caputo formulae and the Grünwald-Letnikov method.

In this paper the authors focus on applying the Riemann-Liouville/Caputo formulae. To investigate a problem of increasing numerical calculations accuracy, the authors apply two methods never used before for such purposes: Gauss-Jacobi Quadrature and Double Exponential Transformation. Additionally application of arbitrary precision instead of double precision to further increase accuracy of two mentioned methods is researched. The Grünwald-Letnikov formula is used for comparison purposes only. A level of accuracy obtained by this formula is treated as one of points of reference.

Despite the most modern computers and comprehensive numerical calculations knowledge, problems connected with difficult kernel integrand included in Riemann-Liouville/Caputo formulae are not solved, because of the singularity at the end point [3–5].

Main motivation for the following research is to solve this difficult computational problem. Second one is to develop a high accurate integration method, which can be applied in a collocation method for solving numerically fractional order differential equations. The increased calculations accuracy enables the authors for more reliable simulations of the close loop dynamic systems control.

Some preliminary high accuracy results proof of the developed methods of integration are presented in appendix to solve integration part of Abel's integral equation of the 1^{st} kind.

2. Mathematical preliminaries

The Riemann-Liouville definition of the fractional integral of order $\alpha > 0$ is given as

$$ {}^{RL}{}_{t_0}I_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} (t-\tau)^{\alpha-1} f(\tau)\, d\tau, \quad (\alpha > 0), \tag{1} $$

where Γ is the gamma function.

Riemann-Liouville Fractional Derivative is defined as

$$ {}^{RL}{}_{t_0}D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^n \int_{t_0}^{t} (t-\tau)^{n-\alpha-1} f(\tau)\, d\tau, \tag{2} $$

Caputo Fractional Derivative is defined as

$$ {}^{C}{}_0D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{0}^{t} (t-\tau)^{n-\alpha-1} f^{(n)}(\tau)\, d\tau, \tag{3} $$

where $\alpha > 0$, n is the smallest integer greater than or equal to α, and the operator $\left(\frac{d}{dt}\right)^n$, $f^{(n)}$ denote the ordinary differential operator (of integer order).

Formulae (2) and (3) are related by the formula

$$ {}^{RL}{}_{t_0}D_t^{\alpha} f(t) = {}^{C}{}_0D_t^{\alpha} f(t) + \sum_{i=0}^{n-1} \frac{t^{i-\alpha}}{\Gamma(i-\alpha+1)} f^{(i)}(0), \tag{4} $$

For the last condition we may switch between the two (2), (3) as necessary [1–3].

*e-mail: dbrzezinski@kis.p.lodz.pl

Grünwald-Letnikov formula of Fractional Order Backward Difference/Sum is defined as the derivative of a real order $\alpha > 0$ (for the integral we use order $-\alpha < 0$) of a continuous bounded function $f(t)$

$$ {}^{GL}\Delta^{\alpha} f(t) = \lim_{h \to 0} \frac{\sum_{i=0}^{\frac{t-t_0}{h}} a_i^{(\alpha)} f(t - hi)}{h^{\alpha}}, \qquad (5) $$

where

$$ a_i^{(\alpha)} = \begin{cases} 1 & \text{for} \quad i = 0 \\ a_{i-1}^{(\alpha)} \left(1 - \dfrac{1+\alpha}{i}\right) & \text{for} \quad i = 1, 2, 3, \ldots \end{cases} \qquad (6) $$

3. Numerical problem formulation and research plan

3.1. Numerical problem in detail. The Kernel integrand in the formulae (1-3) is very difficult to integrate numerically, because of the singularity at the end of an integration range, i.e. the limit of the integrand at the endpoint is infinity (see Fig. 1).

Each time there is a fractional order derivative or integral to compute, no matter which function, order or range, the singularity will badly influence the accuracy level of each, even the most comprehensive method of numerical integration.

Hence the widely applied integration methods are insufficient due to target high precision, especially for orders near 0 (fractional integrals) and 1 (fractional derivatives).

Computations accuracy summary for three methods is presented in Table 1. The methods include Newton-Cotes Midpoint Rule (NCm), Gauss-Kronrod Quadrature (GKr), and Grünwald-Letnikov Fractional Order Backward Difference/Sum (GL).

Table 1
Accuracy expressed as relative error

Fractional derivative of order $\alpha = 0.9$			
N	NCm	GL	GKr
8	6.9e-01	1.6e-01	–
15	–	–	9.0e-01
21	6.2e-01	6.5e-01	9.0e-01
61	5.6e-01	2.3e-03	9.0e-01
600	4.5e-01	2.4e-03	–
Fractional integral of order $\alpha = 0.1$			
N	NCm	GL	GKr
8	6.0e-01	5.7e-03	–
15	–	–	9.1e-01
21	6.3e-01	2.2e-04	9.1e-01
61	5.4e-01	7.5e-04	9.1e-01
600	4.3e-01	7.5e-05	–

Heavside Step Function (23) is the subject of the test for fractional integral and exponential function (22) for fractional derivative. N denotes amount of sampling points.

Table 1 shows, that no applied method can achieve high-accuracy results (GKr' s relative error is 90%). Type of integrated function can additionally decrease or increase final error. The same happens when the order of a computed fractional derivative or integral decreases/increases respectively (it is due to power increase in the denominator of the formulae (1-3)). Only reference method, GL is able to reach accuracy 10^{-04} with 600 coefficients. However, it is difficult for it to reach higher accuracy; to reach 10^{-08} mark, GL requires unimaginable 2.5 billion of coefficients (6). Further Increase of coefficients increases the accuracy even slower.

3.2. Elementary solutions to similar problems. The solution often applied for the singularity problem, transformation of the integrand expression analytically prior the integration is very effective [6], but it is impractical.

Widely known literature of the subject [4, 5 ,7] suggests some strategies when dealing with singularities in numerical integration without analytical pre-treatment, to exclude any singularities from the integration range. In this case, however, the most valuable area of the integrands (1-3) is near singularity; to divide problematic range in smaller parts and apply adaptive strategies, i.e. increase amount of sampling points proportionally to function's values increases for sampling unit; to "stretch" the problematic range, apply more sampling points there and finally, to integrate reciprocal function.

Unfortunately, the magnitude of researched integrand expression difficulty in formulae (1-3) makes all the strategies unhelpful (see Table 1 for details).

3.3. Research plan. High accuracy integration methods require appropriate analytical pre-transformation of the integrand [6]. However, transformation can also be done without an analytical work.

Some of the most efficient and universal forms of an independent variable transformations are based on *Tanh Transformation* [7–13]. They enable to transform each integrand into bell-shaped one (see Fig. 4), which converges very fast. For such re-shaped integrand the application of the trapezoidal rule is advised [8, 14]. This kind of transformation is hereby applied to problematic integrands (see Double Transformation Formula section).

Adopting a weight function of Gaussian Quadrature dedicated to integrands with endpoints singularities, the Gauss-Jacobi Quadrature to overcome the difficulties of formulae (1-3) also guarantees some outstanding results, which are presented (see Gauss-Jacobi Quadrature section).

To improve accuracy of the proposed methods, the authors apply arbitrary precision mathematical libraries GNU GMP/MPFR (see Arbitrary Precision section).

4. Definition of research approach

Below there are presented methods mentioned in research plan.

4.1. Double Exponential Formula. The Double Exponential (DE) formula joins two applied techniques: the DE transfor-

mation applied to the initial integrand and the trapezoidal rule applied to the transformed integrand.

General idea standing behind the DE transformation which was proposed by Schwartz [11] and become known as the Tanh rule (since $x = \tanh(t)$), is as follows:

Let's consider the integral

$$I = \int_a^b f(x)dx, \tag{7}$$

where $f(x)$ is integrable on interval (a, b). The function $f(x)$ may have singularity $x = a$, $x = b$ or at both.

First we apply the following variable transformation

$$x = \phi(t), \quad \phi(-\infty) = a, \quad \phi(\infty) = b,$$

we obtain

$$I = \int_{-\infty}^{\infty} f(\phi(t))\,\phi'(t)dt. \tag{8}$$

Additionally, $\phi(t)$ must be equipped with special property such as $\phi'(t)$ decreases its values to 0 at, at least double exponential rate as $t \to \pm\infty$, i.e.

$$|\phi'(t)| \to \exp(-c\exp(|t|)),$$

where c is some constant. After that, we apply the trapezoidal formula with an equal mesh size to the transformed integrand expression, i.e.

$$I = h\sum_{n=-\infty}^{\infty} f(\phi(nh))\,\phi'(nh),$$

where nh is sampling step.

Truncation of the summation process is done at some $n = -N_-$ and $n = +N_+$, i.e.

$$I_h^N = h\sum_{n=-N_-}^{N_+} f(\phi(nh))\,\phi'(nh), \tag{9}$$
$$N = N_- + N_+ + 1,$$

where N states amount of sampling points of the function.

Since $\phi'(nh)$ as well as the whole expression $f(\phi(nh))\,\phi'(nh)$ converges to 0 at exponential rate at large $|n|$, the quadrature formula (9) is called the Double Exponential [7, 8, 13].

Due to truncation of the summation process (9) at some $n = -N_-$ and $n = +N_+$ function $f(x)$ can have singularities at $x = a$ and/or $x = b$ as long as (7) is integrable over the integration range.

There two kinds of errors should be taken into consideration when implementing the DE formula: discretization error, because we use the trapezoidal rule to approximate an integral and truncation error, because we truncate infinite sum at some N.The optimal strategy is to make both errors equal [7, 8].

The subinterval width$h$, which defines the evaluation step, and the number of sample points are key values in such strategy.

The sources [7, 8] suggest the following value of h for the DE formula

$$h \sim \frac{\log(2\pi N\omega/c)}{N},$$

where c is some constant to be taken, usually 1 or $\pi/2$ and ω is the distance to the nearest singularity of the integrand.

Right selection of a function with optimal properties (10-12) enables to control the level of convergence of the whole transformed expression (9). The rate of convergence has enormous impact on accuracy, i.e. too rapid convergence decreases the accuracy [7, 8].

The authors test three different transformations and select (11) because of its optimal convergence rate for the purpose of the research, which is also suggested by the literature of the subject [7, 13].

The transformation expressions are as follows:

$$\begin{aligned} x &= \phi(t) = \tanh t^p, \\ \phi'(t) &= \frac{pt^{p-1}}{\cosh^2 t^p}, \quad p = 1, 3, 5, \ldots, \end{aligned} \tag{10}$$

$$\begin{aligned} x &= \phi(t) = \tanh(\pi/2\sinh t), \\ \phi'(t) &= \frac{\pi/2\cosh t}{\cosh^2(\pi/2\sinh t)}, \end{aligned} \tag{11}$$

$$\begin{aligned} x &= \phi(t) = \tanh\left(\pi/2\sinh t^3\right), \\ \phi'(t) &= \frac{3\pi/2\cdot t^2\cosh t^3}{\cosh^2(\pi/2\sinh t^2)}. \end{aligned} \tag{12}$$

Applying transformation (11) to the formulae (1-3) according to formula (9), we obtain following trapezoidal form:

$$S = h\sum_{i=1}^{N} f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right)w_i, \tag{13}$$

where

$$x_i = f(\tanh(\pi/2\sinh(t_i)))$$

are nodes and

$$w_i = \frac{\cosh(t_i)}{\cosh^2(\pi/2\sinh t_i)}\cdot\frac{b-a}{2}$$

are weights of the Double Exponential Quadrature;

Additionally: $t_i = -t_a + (i-1)\cdot h$, $i = 0, 1, 2, 3 \ldots N-1$, $h = \dfrac{2t_a}{N-1}$ are the new integration range and width of one trapezoidal panel.

Value of the t_a parameter decides how near the singularity we integrate.

Figures 1–3 visualize the DE transformation: original kernel integrand (1-3), transforming function (11) and final transformed integrand (13), respectively.

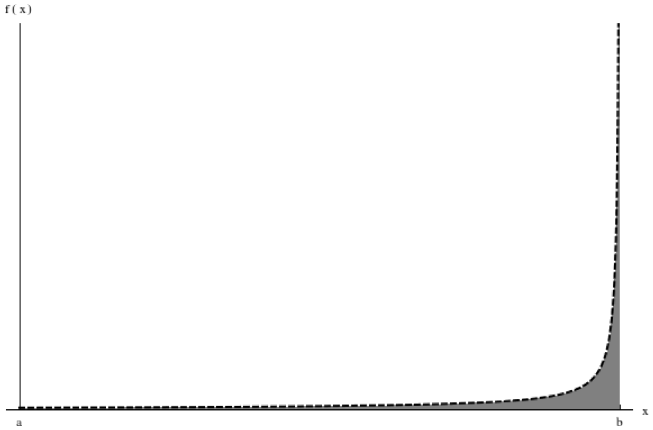

Fig. 1. Graph of the original kernel integrand in formulae (1-3)

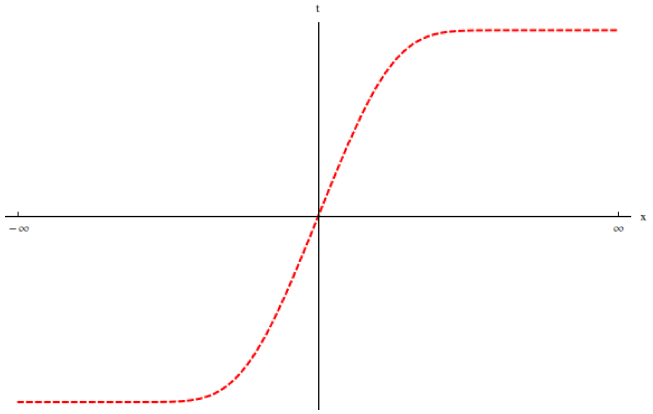

Fig. 2. Graph of transforming function (11)

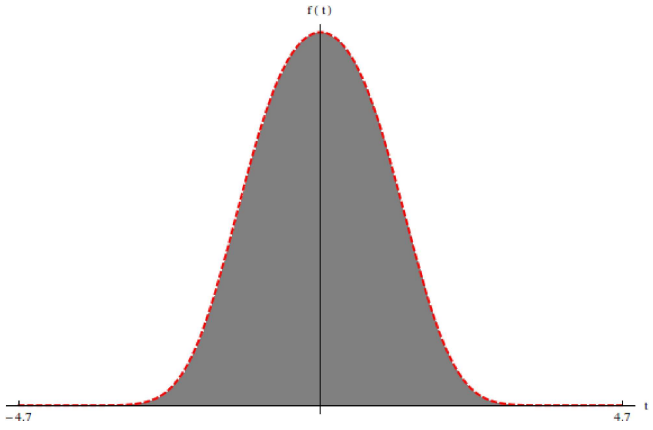

Fig. 3. Graph of the transformed kernel integrand (13) and range applied in computations

4.2. Gauss-Jacobi quadrature. A weight function which allows to eliminate definite integration range endpoints singularities is Jacobi weight (14) [4, 5, 7]

$$p(x) = (1-x)^{\lambda}(1+x)^{\beta}, \quad \lambda, \beta > -1. \tag{14}$$

A Quadrature formula with the weight (14) assumes form

$$\int_{-1}^{1} (1-x)^{\lambda}(1+x)^{\beta} \cdot f(x)\, dx \cong \sum_{k=1}^{n} A_k \cdot f(x_k). \tag{15}$$

The nodes x_k are zeros of Jacobi polynomial $J_n(x;\lambda,\beta)$. With weight (14) Jacobi polynomial is orthogonal to all polynomial of the order lower or equal to $n-1$.

Jacobi polynomial can be determined by applying Rodrigues formula

$$J_n(x;\lambda,\beta) = \frac{(-1)^n}{2^n \cdot n!}(1-x)^{-\lambda}(1+x)^{-\beta} \cdot \frac{d^n}{dx^n}\left[(1-x)^{\lambda+n}(1+x)^{\beta+n}\right]. \tag{16}$$

There are many formulae, which can be applied to compute weight coefficients A_k. For the purpose of numerical calculations, the most convenient is following form [15]

$$A_k = 2^{\lambda+\beta+1}\frac{\Gamma(\lambda+n+1)\Gamma(\beta+n+1)}{n!\Gamma(\lambda+\beta+n+1)} \cdot \frac{1}{(1-x_k^2)\left[J_n^{(\lambda,\beta)'}(x_k)\right]^2}.$$

The reminder of the Gauss-Jacobi Quadrature is expressed as

$$R = \frac{2^{\lambda+\beta+2n+1}}{\lambda+\beta+2n+1} \cdot \frac{\Gamma(\lambda+n+1)\,\Gamma(\beta+n+1)\,\Gamma(\lambda+\beta+n+1)}{\Gamma^2(\lambda+\beta+2n+1)} \cdot \frac{n!}{(2n)!} \cdot f^{(2n)}(\xi), \xi \in \langle -1, 1\rangle.$$

Substituting $\lambda = 1-\alpha$, $\beta = 0$ in the quadrature formula (15), we obtain

$$\int_{-1}^{1} \frac{\phi(x)}{(1-x)^{1-\alpha}} dx, \tag{17}$$

which coincides with the kernel integrand of the formulae (1-3).

To change the integration range from $[-1,1]$ to $[t_0,t]$, formula (17) must be transformed as follows

$$\left(\frac{t-t_0}{2}\right)^{\alpha} \int_{-1}^{1} \frac{\phi(u)}{(1-u)^{1-\alpha}} du, \tag{18}$$

where

$$\phi(u) = f\left(\left(\frac{t-t_0}{2}\right)u + \left(\frac{t+t_0}{2}\right)\right).$$

Applying the formulae (17-18) we can express formula (1) as

$${}^{RL}I^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)}\left(\frac{t-t_0}{2}\right)^{\alpha} \int_{t_0}^{t} \frac{f(u)}{(t-u)^{1-\alpha}} du. \tag{19}$$

To apply the formula (15) to calculate fractional derivatives (2,3) we proceed similar way

$$\left(\frac{t-t_0}{2}\right)^{n-\alpha} \int_{-1}^{1} \frac{\phi(u)}{(1-u)^{n-1-\alpha}} du, \tag{20}$$

where

$$\phi(u) = f\left(\left(\frac{t-t_0}{2}\right)u + \left(\frac{t+t_0}{2}\right)\right).$$

Formula (3) assumes the following form

$$ {}^{C}D^{\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)}\left(\frac{t-t_0}{2}\right)^{1-\alpha} \cdot \int_{t_0}^{t}(t-u)^{n-\alpha-1}\left(\frac{d}{dt}\right)^{n} f(u)du. \tag{21} $$

5. Arbitrary precision

The GNU Multiple Precision Floating-Point Reliable Library (MPFR) is an arbitrary precision package for C language and is based on GNU Multiple-Precision Library (GMP). MPFR supports arbitrary precision floating point variables. It also provides an exact rounding of all implemented operations and mathematical functions [16].

The library enables a user to set the precision of the arbitrary precision variables precisely by specifying the number of bits to use in the mantissa of the floating point number. Due to the design of the library, it is possible to work with any precision between 2 bits and maximum bits allowed for a computer. The ability of MPFR to set the precision exactly to the desired precision in bits is the major difference of this library in comparison with competitors. It is also the reason for choosing it for the following investigation.

The most common errors in numerical calculations are caused by wrong rounding. The MPFR library supports exact rounding in compliance with IEEE 754-2008 standard. It implements four of the rounding modes specified by the standard, as well as one additional not included in it:

1. Rounding to nearest, ties to even
2. Round toward 0
3. Round toward $+\infty$
4. Round toward $-\infty$
5. Round away form 0 (not in the IEEE 754-2008 standard).

6. Numerical accuracy analysis

6.1. Integer order integration case. The purpose of the following test is to compare the accuracy of FOD/I numerical calculations applying two proposed methods:

- Double Exponential Quadrature (DE),
- Gauss-Jacobi Quadrature (GJ),

with:

- Newton-Cotes Midpoint Rule (NCm) (modification of Rectagular Rule),
- Gauss-Kronord Quadrature (GKr) (modification of Gauss-Legendre Quadrature),
- Grünwald-Letnikov (GL) formula of Fractional Order Backward Difference/Sum (4).

GKr is based on the Gauss-Legendre Rule. The G7/K15, so called Gauss-Kronrod Pair includes the nodes of the 7-point Gauss-Legendre Quadrature +8 new ones and all 15 new coefficients [7]. Midpoint Rule (NCm) is a modification of standard Rectangular Rule, which enables the application of it to the integrands with singularities, i.e. the sampling points are taken in the middle of each interval.

Grünwald-Letnikov (GL) is already presented. It is the only method dedicated to FOD/I computations.

The algorithms of the methods applied in the comparison can be distinguished in one main way: DE, NCm and GL use equally spaced sampling points (coefficients in case of GL). GJ and GKr use n^{th} order of appropriate polynomial in approximation; sample points are zeros of the polynomial. Therefore it is difficult to compare the accuracy of these methods with the same amount of sample points. Instead, we can ascertain how many sample points is necessary to achieve target accuracy. In this way it is easier to find out how many of sample points should be set for each method during the main, fractional order comparison.

There is calculated 1^{st} order integral of exponential function e^{at}, $a = 1$, $t \in (0, 1)$ using double precision. The calculations results are presented in Fig. 4. As accuracy criterion relative error e_r is assumed $e_r = 1 - \frac{\nu_c}{\nu_e}$, in which: ν_e is exact value and ν_c is calculated value.

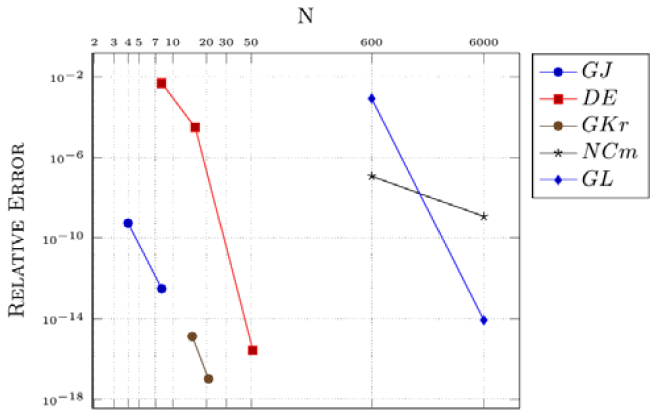

Fig. 4. Accuracy of 1^{st} order integration

As it is observed in Fig. 4, it is possible to achieve high accuracy applying GJ, DE and GKr with over a dozen of sample points. However, in case of NCm and GL, to achieve similar accuracy, it is necessary to apply some several thousand of them [17].

6.2. Fractional order integration case. The amount of sample points for each method during the fractional order comparison is decided on the basis of integral order integration case results:

1. NCm: 600 and 6000 sampling points,
2. GL: 600 and 6000 coefficients,
3. GKr: with pairs 7/15, 10/21 and 30/61,
4. DE: 4-52 (with step 4),
5. GJ: 1-12 (with step 1), order of polynomial applied in approximation.

There are some elementary functions selected:

$$ f(t) = e^{at}, \quad a = 2, \quad t \in (0, 1), \tag{22} $$

$$ f(t) = 1(t), \quad t \in (0, 1), \tag{23} $$

$$f(t) = \sin(t), \quad t \in (0, 1). \tag{24}$$

The integrand in formulae (1-3) is a product of two functions: integrated function (22-24) and kernel of Riemann-Liouville/Caputo formulae.

To illustrate the efficiency and remarkable accuracy of the proposed methods (GJ and DE), the authors stress in the test the following "difficult" orders of fractional derivatives and integrals: 0.0001, 0.001, 0.01, 0.1, 0.5, 0.9, 0.99, 0.999, 0.9999. In all of the tests, the same procedure is applied.

As the final part of the test, the authors compare the accuracy obtained applying double and arbitrary precision. To complete this task two versions of the C++ code are prepared for each proposed method of numerical integration. Arbitrary precision code involved 100 significant digits is compared to accuracy achieved with standard 16 significant digits available in double precision.

7. Assumed exact values

Generally there are no analytic formulae to compute exact values of FOD/I for the purpose of the accuracy estimation as it is in the case of computing classical first derivative or the integral of the 1^{st} order.

There are however computational only formulae available in the literature of the subject [1–3] instead. They are calculated using computer and therefore, there must be taken into consideration some calculations error, although very small.

Assuming $D^{-\alpha} = I^{\alpha}$.
For function (22):

$${}_{t_0}D_t^{-\alpha} f(t) = t^{\alpha} \sum_{k=0}^{N} \frac{(at)^k}{\Gamma(k+1+\alpha)}, \tag{25}$$

$${}_{t_0}D_t^{\alpha} f(t) = t^{-\alpha} \sum_{k=0}^{N} \frac{(at)^k}{\Gamma(k+1-\alpha)}. \tag{26}$$

For function (23)

$${}_{t_0}D_t^{\pm\alpha} f(t) = \frac{t^{\mp\alpha}}{\Gamma(1 \mp \alpha)}. \tag{27}$$

For function (24):

$${}_{t_0}D_t^{-\alpha} f(t) = t^{\alpha} \sum_{k=0}^{N} \frac{(at)^k \sin(0.5k\pi)}{\Gamma(k+1+\alpha)}, \tag{28}$$

$${}_{t_0}D_t^{\alpha} f(t) = \frac{1}{t^{\alpha}} \sum_{k=0}^{N} \frac{(at)^k \sin(0.5k\pi)}{\Gamma(k+1-\alpha)}, \tag{29}$$

where N is arbitrary chosen number.

Additionally, key values were also confirmed comparing the value of the classical first derivative and the integral of the 1^{st} order as well as the value of the function applying fractional order differentiation and integration operators concatenation and fractional order differentiation and integration of the Mittag-Leffler special function [1–3].

8. Error definition

For the purpose of the following comparison analysis, there are calculated derivatives and integrals of selected functions (22-24) in the $(0, 1)$ range of the fractional orders $\langle 0, 1\rangle$ with step 0.1. Additionally derivatives and integrals of "difficult" orders as for example 0.001 and 0.999 are calculated to test the developed numerical integration methods abilities in the most difficult conditions, in which available methods of numerical integration deliver results usually burdened with 100–200% of relative error.

The criterion of computational accuracy and the only criterion of the usefulness for the purpose of the research of the developed methods of integration is assumed to be relative error e_r expressed as

$$e_r = 1 - \frac{\nu_c}{v_e},$$

where v_e is a value assumed as exact obtained by calculated applying formulae (25-29) and v_c is calculated value applying developed methods of integration.

Fractional (non-integer) order derivatives unlike classical, integer order derivatives are non local values, i.e. they are calculated using the information from the whole range of integration. It is worth explaining, that even if the selected range for the calculations is $(0, 1)$, the value of the calculated derivative concerns always the end of the integration range, i.e. in this case 1 (results in Figs. 5–9, 12, 16, 17). The presented results in the following comparison are, however valid for any other ranges.

Efficiency is the next part of the enclosed results presented as dependency of computational accuracy (expressed as relative error) on amount of sampling points (order of polynomial) applied in calculations for the same method (results in Figs. 10–11).

Computational complexity is not the subject of the following research at the present stage of algorithms development. Additionally, due to large variation in amount of mathematical operations required by each of the tested integration methods, the authors decide to present at the moment only estimated computational complexity in form of roughly estimated computational time complexity, i.e. the computational time required to deliver results with desired accuracy by each of the methods. The charts in the this part of the comparison (in Figs. 14, 15) present the arithmetic mean value of the computing time calculated out of 1000 the same calculations for each method and each function.

9. Results for double precision

The first part of the results presentation (see Figs. 5–9 and 12) concerns the evaluation of applied methods of numerical integration in context of possible accuracy of computations:

- Generally applying GJ ensures steady and predictable highest accuracy of all methods (full double precision range) for all ranges and orders of fractional derivatives and integrals. Wherein, only 1–8^{th} order of polynomial application

was necessary to apply, to obtain such remarkable high accuracy.

- Accuracy of the DE depends on order of FOD/I which is to calculate. It offers satisfactory results (half of the double precision range) only for half of tested orders. The DE method brings higher accuracy than GJ only in high orders of fractional integrals and low orders of fractional derivatives. Unfortunately, increasing with order power in denominator in formulae (1-3) and limitation of the transforming function (11) cripples the exceptional abilities of this, otherwise interesting method (see Fig. 13 for details). As it is observed in Fig. 12, The DE method application can increase possible accuracy over GJ by 3-4 orders. However, with slightly more sampling points (42-56).
- GL offers steady accuracy on the level of 10^{-04}. No surprise here [15].
- Other applied methods' results are not satisfactory at all; relative error increases squarely as the order of fractional derivative approaches 1 and fractional integral approaches 0.

Let's focus now on the most accurate and most efficient method, GJ:

- It's exceptional efficiency is presented in the ultimate accuracy test, to calculate fractional integrals of orders near 0 (and fractional derivatives near 1) (see Figs. 10 and 11 details).
- As is it observed in Figs. 5–9 and 12, the GJ method achieves highest accuracy for all orders; additionally the method also guarantees highest results for "difficult" orders 0.1 to 0.0001, for which other methods can't even reach one digit relative error; GJ achieves it with only 1-6th order of polynomial applied (denoted as N on the charts).
- Applying GJ relative error decreases in the same situation, when the error increases applying the other methdos.

To present superiority of proposed numerical methods of integration, the authors decided to conduct their preliminary calculations time complexity test. Preliminary, because, as it was stated earlier, the algorithms are still in the development stage.

The results for this part of the evaluation conducted applying Linux *time* utility for example function (22) are presented in Fig. 14. The chart presents running time against accuracy (relative error) possible to achieve by GJ, DE and GL. Comparison test was conducted on a standard PC equipped with Intel Core 2 Duo processor with 2.2Ghz each core and 4 GB of memory:

- GJ is slightly more time consuming (a few ten-thousandth of a second) than GL and DE, but increasing the order of polynomial does not affect the required running time, i.e. the method guarantees the highest accuracy without extend calculation time.
- GL offers quick calculation times, but only for the accuracy 10^{-04}. Higher accuracy requires almost linear calculation time increase.
- Increasing target accuracy does also not affect DE in context of running time.

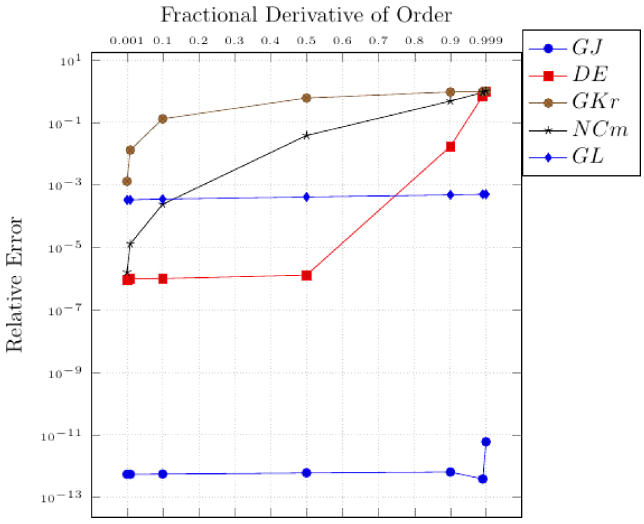

Fig. 5. Calculations accuracy for function (22)

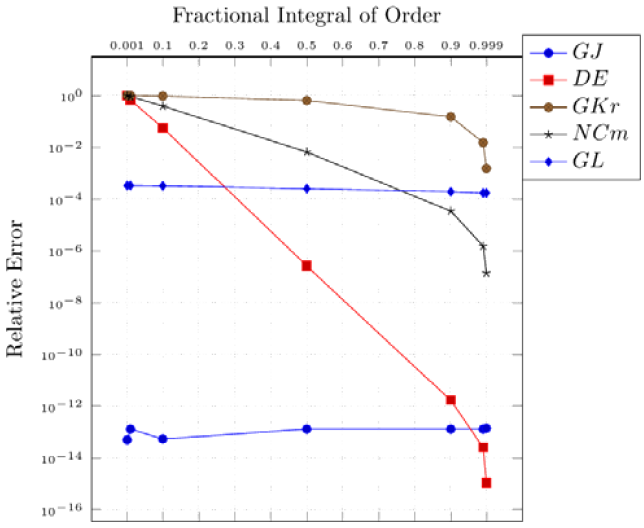

Fig. 6. Calculations accuracy for function (22)

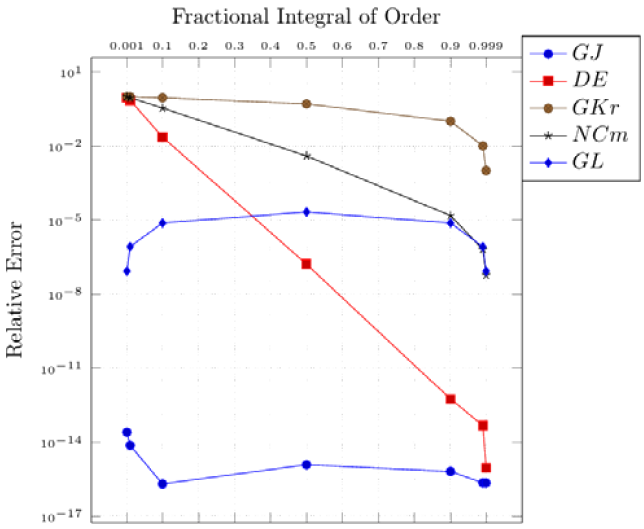

Fig. 7. Calculations accuracy for function (23)

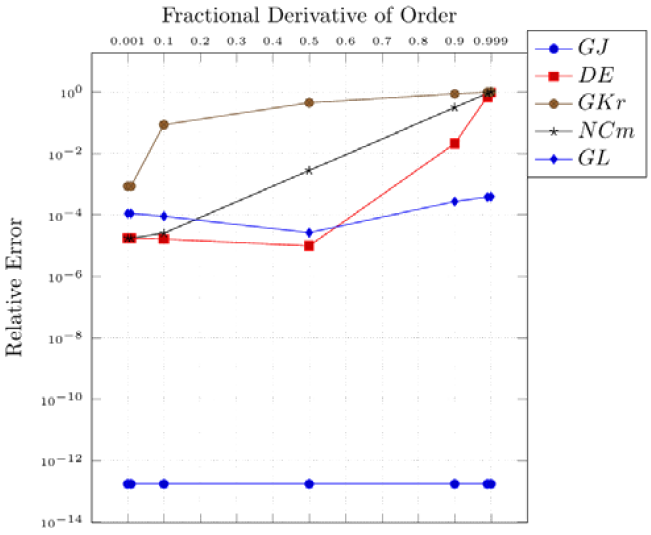

Fig. 8. Calculations accuracy for function (24)

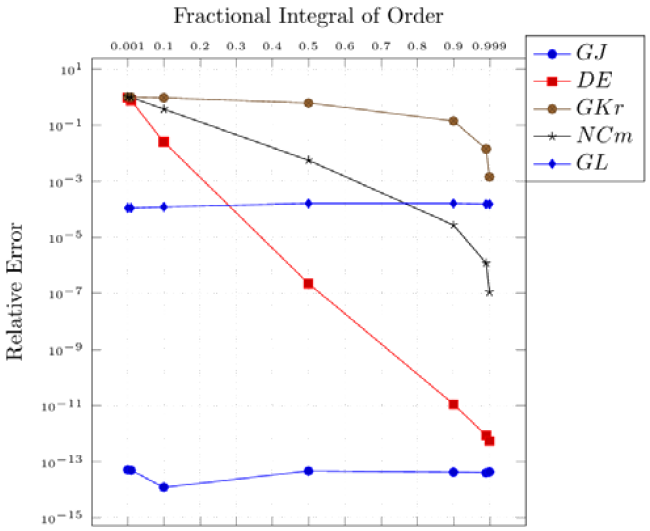

Fig. 9. Calculations accuracy for function (24)

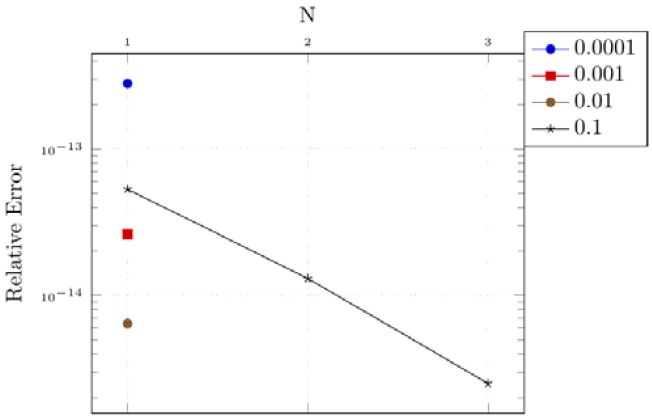

Fig. 10. Calculations accuracy of GJ for fractional integrals (24)

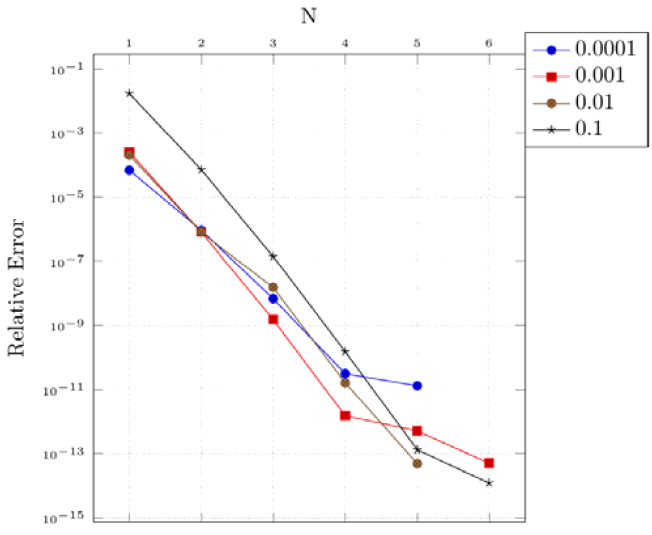

Fig. 11. Calculations accuracy of GJ for fractional integrals (22)

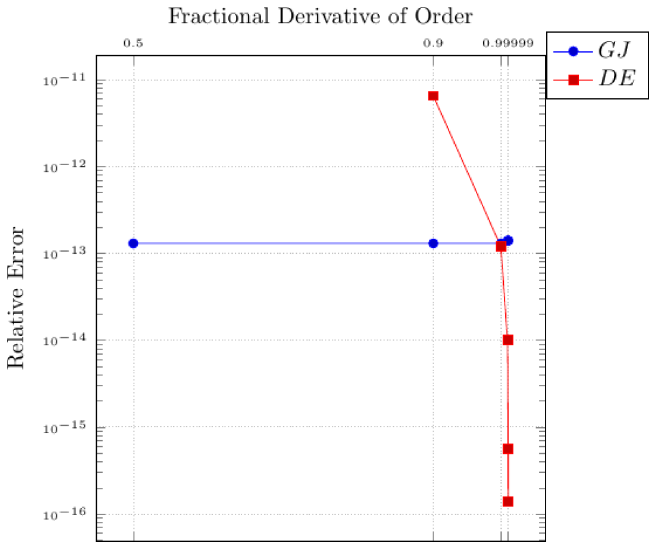

Fig. 12. Calculations accuracy for function (22)

10. Results for arbitrary precision

Careful analysis of the Figs. 16, 17 suggests, that outstanding results of an arbitrary precision application over standard, double precision should be emphasized:

- Arbitrary precision enables to calculate more accurately components of each of the applied methods. More accurately operated methods of integrations result in obtaining more accurate measurements.
- For example, Gaussian types quadratures are characterized by sampling points, which are the zeros of corresponding polynomials. If they are very near to each other and the calculations precision is restricted to standard 16 significant digits, they can appear for calculations as the same. Application of arbitrary precision (in this case 100 significant digits, 300 bits for mantissa of each floating point number) helps to avoid such restrictions and increase overall accuracy.

- Second example is the parameter t_a from (13) in the DE method, which controls how near to the singularity we integrate. With the help of increased precision, there is naturally, possible to integrate nearer the singularity. This, of course, increases the accuracy of this method (see Fig. 17).

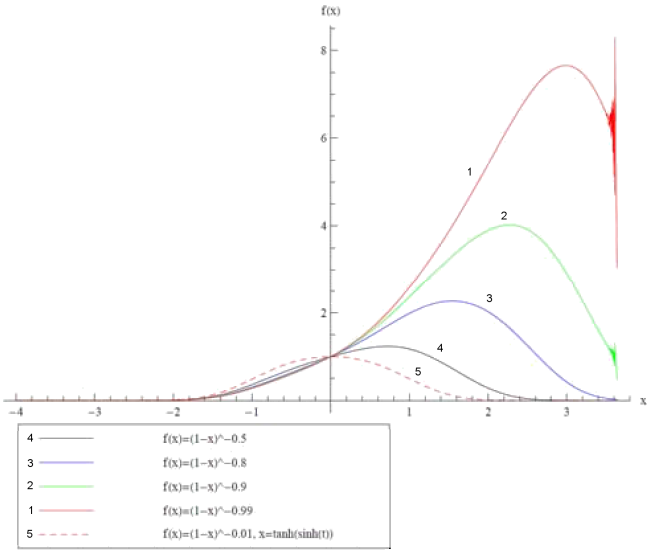

Fig. 13. Abilities of the DE (13) formula crippled by increasing power of the denominator in the formulae (1-3)

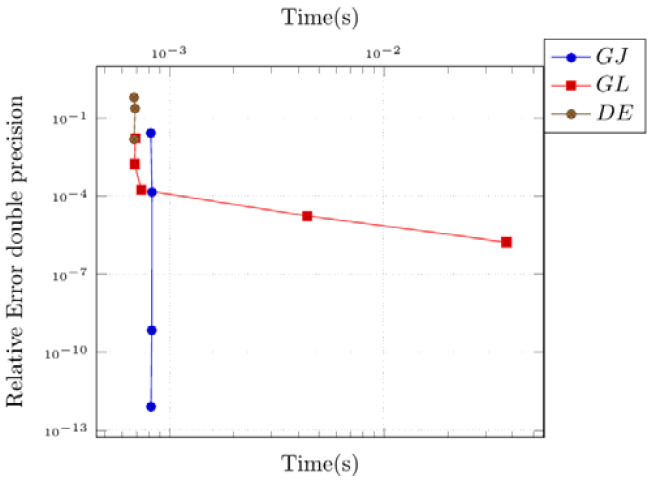

Fig. 14. Programs running time for function (22)

Summarizing, application of arbitrary precision in form of GNU GMP and GNU MPFR arbitrary precision and correct rounding mathematical libraries helps to overcome many technical problems common for double precision and considerably increase the accuracy of computations.

Accuracy comparison is presented in Fig. 16 (for double precision) and Fig. 17 (for arbitrary precision).

For calculations time complexity of the arbitrary precision, the situation is similar as observed for applied double precision (with proportionally longer calculation times due to six folded increased accuracy in comparison) (see Fig. 15 for details).

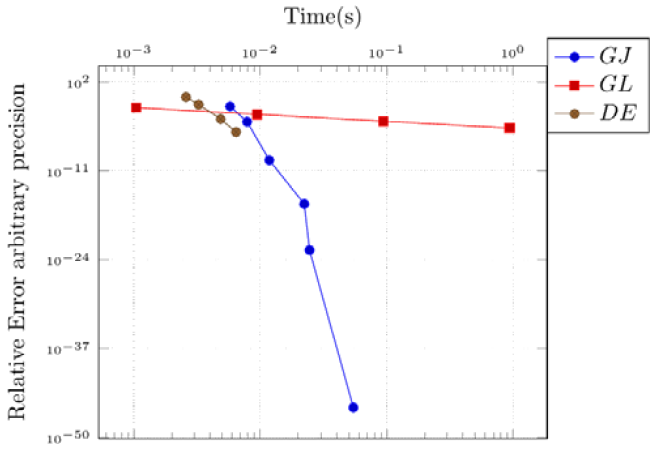

Fig. 15. Programs running time for function (22)

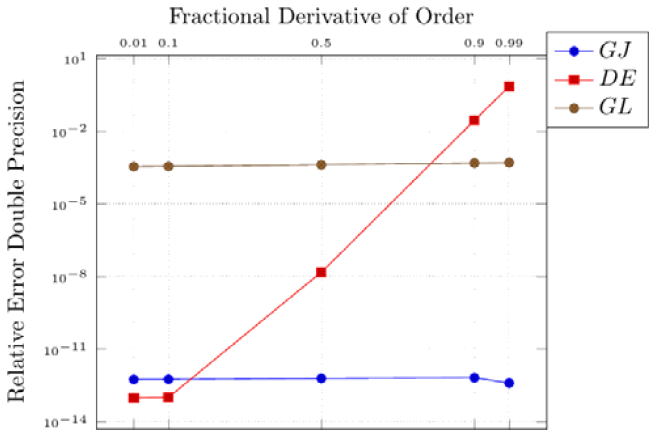

Fig. 16. Double precision results for function (22)

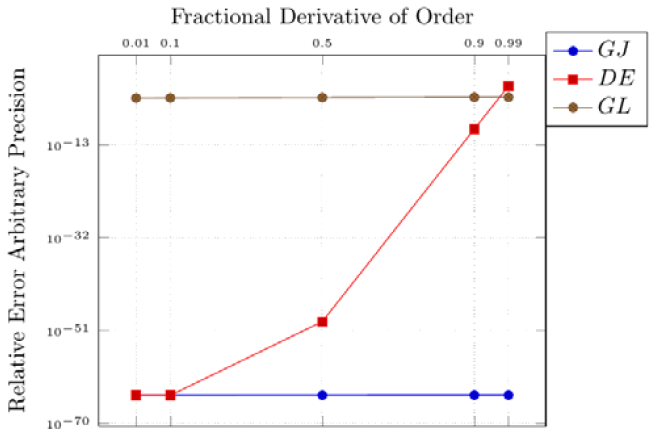

Fig. 17. Arbitrary precision results for function (22)

11. Conclusions

The Gauss-Jacobi Quadrature (GJ) with an adapted weight function delivers steady, remarkable, high accuracy results for all fractional orders and ranges. The application of the Double Exponential Quadrature (DE) in a narrow range of orders, increases the accuracy gain obtained by GJ.

The application of GNU GMP/MPFR arbitrary precision libraries, doubles accuracy gain of both methods.

All methods can be used as an replacement for the Grünwald-Letnikov method, which are usually applied in practical technical applications [18–21], because they deliver steady unmatched accuracy for all orders and ranges with a remarkably low amount of sample points (6-8 instead of some hundred) and steady, short calculation times. The authors also plans to apply the methods to solve diffusion-wave equation with higher accuracy, inspired by existing works [22–24].

Appendix

Abel's integral equation. Many problems in engineering can be solved by the Volterra integral equation of the first kind

$$\int_0^t k(t,s)\phi(s)ds = f(t),$$

where $f(t)$, $k(t,s)$ are given functions and $\phi(t)$ is an unknown function. The function $k(t,s)$ is the kernel of this integral equation. There is a special case of this kernel, which is the linear Abel operator

$$J_\nu\phi(t) = \frac{1}{\Gamma(\nu)}\int_0^t (t-s)^{\nu-1}k(t,s)\phi(s)ds,$$
$$0 \le t \le 1, \qquad 0 < \nu < 1.$$

In the case $k(t,s) = 1$, the operator J_ν is the classical Abel operator.

The solution of the classical Abel integral equation

$$\frac{1}{\Gamma(\nu)}\int_a^x \frac{g(t)}{(x-t)^\nu}dt = f(x), \quad 0 < \nu < 1, \qquad a < x < b$$

is given by

$$g(x) = \frac{1}{\Gamma(1-\nu)}\frac{d}{dx}\int_0^x (x-u)^{\nu-1}f(u)du.$$

There is analogy between Abel's integral equation fractional order integral of a function. In fact, the fractional integral of order ν of a function $f(x)$ is just Abel operator $J_\nu f(t)$.

The operator $D^\nu f(x) = \dfrac{d}{dx}J^{1-\alpha}f(x)$, which in fact is (2).

If $f(t) = e^{-0.5t}$, $0 < t < 1$, then

$$J^\nu f(x) =_0 D_t^\nu E_{\alpha,\beta}(z),$$
$$z = az,$$
$$\alpha = 1,\ \beta = 1,\ t = 1,\ a = -0.5.$$

where $E_{\alpha,\beta}(z)$ is Mittag-Leffler function.

In the Table 2 there are presented evaluation results of $J^\nu f(x)$ for $\nu = 0.15$, $v = 0.75$ applying Gauss-Jacobi Quadrature with adopted weight function ($N = 16$), Double Exponential Transformation (DE) ($N = 600$) and reference method, Grünwald-Letnikov Backward Difference/Sum (GL) ($N = 600$).

Table 2
Accuracy comparison among applied methods to calculate $J^\nu f(x)$

$\nu = 0.15$	Relative Error		
x	GJ	DE	GL
0.1	5.8e-56	3.3e-15	3.4e-05
0.2	1.5e-53	3.3e-15	3.8e-05
0.3	1.8e-53	3.2e-15	1.1e-04
0.4	2.7e-54	3.2e-15	1.8e-04
0.5	1.8e-53	3.2e-15	2.5e-04
0.6	5.0e-53	3.2e-15	3.2e-04
0.7	7.6e-53	3.1e-15	4.0e-04
0.8	7.6e-53	3.1e-15	4.7e-04
0.9	5.6e-53	3.1e-15	5.4e-04
1.0	3.2e-53	3.1e-15	6.1e-04
$\nu = 0.75$	Relative Error		
x	GJ	DE	GL
1.0	4.0e-54	3.1e-54	1.1e-04
0.9	2.0e-54	5.6e-54	6.1e-05
0.8	2.2e-54	7.9e-54	1.4e-05
0.7	4.9e-53	8.2e-54	3.3e-05
0.6	2.1e-54	5.5e-54	8.1e-05
0.5	5.5e-54	2.1e-54	1.3e-04
0.4	8.2e-54	4.9e-54	1.7e-04
0.3	7.9e-54	2.2e-54	2.2e-04
0.2	5.6e-54	2.0e-54	2.7e-04
0.1	3.1e-54	4.0e-54	3.2e-04

REFERENCES

[1] R. Gorenflo and F. Mainardi, "Fractional calculus: integral and differential equations of fractional order", in *Fractals and Fractional Calculus in Continuum Mechanics*, Springer Verlag, Wien, 1997.

[2] I. Podlubny, *Fractional Differential Equations. Mathematics in Science and Engineering*. vol. 198, Academic Press, New York, 1999.

[3] D. Baleanu, K. Diethelm, E. Scalas, and J.J. Trujillo, *Fractional Calculus Models and Numerical Methods*, World Scientific Pub. Co. Inc, London, 2012.

[4] P.K. Kythe and M.R. Schäferkotter, *Handbook of Computational Methods for Integration*, Chapman & Hall/CRC, London, 2005.

[5] R.L. Burden and J.D. Faires, *Numerical Analysis*, Brroks/Cole Cengage Learning, Boston, 2003.

[6] D.W. Brzeziński and P. Ostalczyk, "Numerical evaluation of fractional differ-integrals of some periodical functions via the IMT transformation", *Bull. Pol. Ac.: Tech.* 60 (2), 285–292 (2012).

[7] W.H. Press, S.A. Teukolsky, and B.P. Vetterling, *Numerical Recipes. The Art of Scientific Computing*, Cambridge University Press, Cambridge, 2007.

[8] M. Mori, "Developments in the double exponential formulas for numerical integration", *Proc. Int. Congress of Mathematicians* 1, CD-ROM (1990).

[9] N.M. Korobov, *Teoretikocislowyie Metody w Priblizennom Analize*, Gosudarstwennoe Izdatelstwo Fizyko-Matematiceskoj Literatury, Moscow, 1963.

[10] S. Haber, "The Tanh rule for numerical integration", *SIAM J. Numer. Anal*. 14, 668–685 (1999).

[11] C. Schwartz, "Numerical integration of analytic functions", *J. Computational Physics* 4, 19–29 (2000).

[12] F. Stenger, "Integration formulae based on the trapezoidal formula", *J. Inst. Math. Appl.* 12, 14 (1973).

[13] H. Takahasi, *Quadrature Formulas Obtained by Variable Transformation*, Springer Verlag, Berlin, 1973.

[14] J. Waldvogel, "Towards a general error theory of the trapezoidal rule", in *Approximation and Computation* 42, 267–282 (2011).

[15] N. Hale and A. Townsend, "Fast and accurate computation of gauss-legendre and gauss-jacobi quadrature nodes and weights", *Siam J. Sci. Comput*. 35, 652–674 (2013).

[16] K.R. Ghazi, V. Lefèvre, P. Théveny, and P. Zimmermann, "Why and how to use arbitrary precision", *IEEE Computer Society* 12, 5 (2010), DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/MCSE.2010.73.

[17] D.W. Brzeziński and P. Ostalczyk, "The Grünwald-Letnikov formula and its horner's equivalent form accuracy comparison and evaluation to fractional order PID controller", *IEEE Explorer Digital Library: IEEE Conference Publications*, 17^{th} *Int. Conf. Methods and Models in Automation and Robotics* MMAR'12, CD-ROM (2012).

[18] M. Busłowicz and A. Ruszewski, "Necessary and sufficient conditions for stability of fractional discrete-time linear state-space systems", *Bull. Pol. Ac.: Tech.* 61 (4), 779–786 (2013).

[19] T. Kaczorek, "Decoupling zeros of positive continuous-time linear systems", *Bull. Pol. Ac.: Tech.* 61 (3), 557–562 (2013).

[20] W. Mitkowski and P. Skruch, "Fractional-order models of the supercapacitors in the form of RC ladder networks", *Bull. Pol. Ac.: Tech.* 61 (3), 581–587 (2013).

[21] A. Dzielinski, G. Sarwas, and D. Sierociuk, "Comparison and validation of integer and fractional order ultracapacitor models", *Advances in Difference Equations* 1, 11–23 (2011).

[22] W. Mitkowski, "Approximation of fractional diffusion-wave equation", *Acta Mechanica et Automatica* 5, 65–68 (2011).

[23] A. Obrączka and W. Mitkowski, "The comparison of parameter identification methods for fractional partial differential equation", *Diffusion and Defect Data – Solid State Data. Part B, Solid State Phenomena* B 210, 265–270 (2014).

[24] M. Błasik and M. Klimek, "Exact solution of two-term nonlinear fractional differential equation with sequential Riemann-Liouville derivatives", *Advances in the Theory and Applications of Non-integer Order Systems. Lecture Notes in Electrical Engineering* 257, 161–170 (2013).