Int. J. Dynam. Control (2017) 5:18–29
DOI 10.1007/s40435-016-0236-z

# Trapezoidal rule and its error analysis for the Grünwald-Letnikov operator

**Matthew Harker**[1] · **Paul O'Leary**[1]

Received: 1 December 2015 / Revised: 11 February 2016 / Accepted: 16 February 2016 / Published online: 11 March 2016

**Abstract** In this paper, the trapezoidal rule for the Grünwald-Letnikov operator is derived. It is a trapezoidal rule in the sense that the formula yields the exact Grünwald-Letnikov derivative/integral of a piecewise linear function. Firstly, the formula for evenly spaced points is derived, and is used as a basis to derive the equivalent formula for arbitrary abscissae. Further, an analytic bound on the residual error is derived, which depends on a bound on the second derivative of the function. The derived trapezoidal rule can therefore be used to compute fractional integrals and derivatives to within a given error tolerance. Through numerical testing it is shown that the new formula yields results that are orders-of-magnitude more accurate than the classical formula, even for arbitrary functions. A simple adaptive algorithm is proposed for computing the result of applying the Grünwald-Letnikov operator to a function to within a desired accuracy.

**Keywords** Grünwald-Letnikov operator · Trapezoidal rule · Error analysis

## 1 Introduction

The numerical calculation of fractional order derivatives and integrals is critical to the simulation of fractional order systems. Specifically, a good approximation is essential to the numerical solution of fractional order differential equations [1,2], as well as partial fractional order differential equations [3,4]. One of the more important operators in Fractional Calculus is the Grünwald-Letnikov operator [5–7] of arbitrary order $\alpha$, since it encompasses both the Riemann integral as the special case $\alpha = -1$ as well as the definition of the derivative as the limit of a ratio of differences as the special case $\alpha = 1$. In terms of numerical algorithms, much attention has been given to specific fractional derivatives, such as the Caputo derivative [8,9], and the Riemann-Liouville derivative [9,10], including some higher order formulas. Strangely, relatively little attention has been given to the actual numerical computation of the Grünwald-Letnikov operator, in spite of the fact that its classical approximation appears to be the most commonly used formula for the numerical computation of fractional order derivatives. The classic approach to computing Grünwald-Letnikov derivatives and integrals is to take the definition of the operator at a point $x > 0$, i.e.,

$$
{}_0D_x^{\alpha} f(x) = \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{x}{n}\right)^{(-\alpha)} \sum_{k=0}^{n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)}\, f\left(x - k\frac{x}{n}\right) \qquad (1)
$$

where $\Gamma(z)$ is the Gamma Function, and to simply take a finite value for $n$, equal to the number of abscissae. This yields the coefficients of a linear combination of function values for the approximation. This particular approach is advocated in the majority of the literature, e.g., [6,7,10]. The formula has the advantage that it can be computed relatively simply using the recurrence formula for the Gamma function [7]. This approach is prevalent enough that there have been recent analyses as to the accuracy of this classic formula [11]. There are, however, two significant disadvantages due to the nature of the formula: Firstly, since the formula is only exact in the limit $n \to \infty$, it only yields accurate results for a very large number of abscissae; secondly, the coefficients only make sense when the abscissae are evenly spaced, limiting the use

✉ Matthew Harker
matthew.harker@unileoben.ac.at

[1] Institute for Automation, University of Leoben,
Peter-Tunner-Strasse 27, 8700 Leoben, Austria

of the approximation to evenly spaced function evaluations. In practical terms, this means that more often than not, a large amount of computation must be undertaken in order to yield acceptable results. An alternative approximation is given by Podlubny et al. [12]; their approximation for differentiation involves the matrix inversion of an integration operator, which unfortunately can lead to instability for a larger number of abscissae.

The new approach taken in this paper is to compute the necessary coefficients in order to yield the exact result of the Grünwald-Letnikov operator applied to a piecewise linear function. In the case of $\alpha = -1$, this approach is the well known trapezoidal rule for numerical integration. In fact, the generalization of the trapezoidal rule derived in this paper yields exactly the important limit cases: The trapezoidal rule for integration, when $\alpha = -1$; the appropriate identity operator for $\alpha = 0$; and finally, the backwards difference operator for the case $\alpha = 1$. Given that the new approximation is exact for piecewise linear functions over arbitrary spacing, it follows that the new method can be used to obtain more accurate solutions to fractional differential equations than the classic method with far less computational work. The method is also effective for large spacing, which is elusive to the classic method.

In order that the proposed trapezoidal rule can be used as a veritable theorem of numerical analysis, we derive an analytic bound on the residual error of the approximation. Specifically, if the second derivative of the function to which the trapezoidal rule is applied can be bounded, then the residual error of the computation can be bounded. The bound is valid provided that $f''(x)$ is continuous, and that $\alpha \leq 1$. The bound therefore covers all of integration, i.e., $\alpha \leq 0$, as well as differentiation for $\alpha$ in the range $0 \leq \alpha \leq 1$. The case $\alpha = 1$ corresponds to backward differences, which are known to have low accuracy for computing derivatives, especially when spacing $h$ is not chosen with care [13]. Further, the case $\alpha = 2$ corresponds to a second derivative, whereby, the second derivative of a piecewise linear function vanishes everywhere. It can therefore be said, that if a practitioner wishes to compute a derivative of order $1 < \alpha < 2$, a higher order formula should be used. The error bound, which is valid for $\alpha < 1$, can be used for two immediate applications: Firstly, the bound can be used to calculate the numerical result to within a given error tolerance; an example of this is given in the present paper; secondly, the error bound can be used to justify the short-memory principle for numerical computation over very large data sets [7]. Previously this approach has been justified only by specific examples; the proposed error bound would give a precise bound for the general case.

Another aspect of the proposed method is the matrix-based approach. Since integration and differentiation are linear operators, it is not surprising that standard methods for approximating integrals and derivatives are to determine an appropriate linear combination of a finite number of function values. In particular, fractional order operators often require a linear combination of all function values. To this end, a matrix based approach to numerical Fractional Calculus greatly simplifies results, and leads to more accurate solutions (Cf. [2,3]). Specifically, it is known that the classic Grünwald-Letnikov approximation is a Toeplitz matrix [2,9]. Toeplitz matrices have a number of special properties which can simplify computation [14]. It is shown in this paper that the same advantage holds for trapezoidal rule for the Grünwald-Letnikov operator since for evenly spaced points it can be written as a sum of Toeplitz matrices. In general, throughout the paper we derive formulæ for the approximation of a linear operator over a partition of an interval, namely, evaluation at each of the points $x_1, x_2, \ldots, x_n$. It is assumed that the lower limit of the Grünwald-Letnikov operator is $x_1$, i.e., the first node, whereby in the case of even spacing it suffices to assume it to be zero. Firstly, we derive the operator for the special case of evenly spaced abscissae. This provides us with sufficient groundwork to approach the case of arbitrary spacing. We thereby follow with a derivation of the general trapezoidal rule for the Grünwald-Letnikov operator, indicating the key steps in the derivation. Furthermore, we provide MATLAB® code for the implementation of the evenly spaced case for both the trapezoidal rule, and the proposed error bound matrix, in order that the formulæ can be independently verified. By means of extensive numerical testing, we demonstrate that the newly derived formulæ are indeed exact for piecewise linear functions, and can be used for accurate computation of fractional derivatives and integrals of arbitrary functions.

## 2 Evenly spaced abscissae

Since we are interested in computing the value of the Grünwald-Letnikov operator applied to a piecewise linear function, it suffices to consider its value at an arbitrary index $i + 1$. In order to derive the trapezoidal rule, we therefore recall the formula for the Grünwald-Letnikov operator (1) of an arbitrary function evaluated at the point $x_{i+1}$ with $x_1 = 0$, i.e.,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) = \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{x_{i+1}}{n}\right)^{(-\alpha)} \times \sum_{k=0}^{n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f\left(x_{i+1} - k\frac{x_{i+1}}{n}\right). \tag{2}
$$

Since we are interested in asymptotically large values of $n$, we may divide each interval into $n$ points, yielding a total of $in + 1$ points. This yields the formula,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) = \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{x_{i+1}}{in}\right)^{(-\alpha)} \times \sum_{k=0}^{in-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f\left(x_{i+1} - k\frac{x_{i+1}}{in}\right). \tag{3}
$$

For evenly spaced points with spacing $h$ and lower limit $x_1 = 0$, we have $x_{i+1} = ih$; thereby, partitioning the summation over each interval, we have,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) = \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{h}{n}\right)^{(-\alpha)} \times \sum_{j=0}^{i-1} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f\left(x_{i+1} - k\frac{x_{i+1}}{in}\right). \tag{4}
$$

Each term in the summation over $j$ now corresponds to a single interval, namely, the interval $[x_{i-j}, x_{i-j+1}]$. It should be noted that this remains an *exact* expression for the Grünwald-Letnikov operator. For a piecewise linear approximation to the function $f(x)$, the function is approximated by a straight line over each interval. This can be accomplished best for our purposes by defining the piecewise linear function,

$$
g(x) = g_{i-j}(x) \quad \text{for} \quad x \in [x_{i-j}, x_{i-j+1}], \tag{5}
$$

with $j = 0, \ldots, i-1$, with each segment defined by the Lagrange interpolating polynomial,

$$
g_{i-j}(x) = \left(\frac{x - x_{i-j+1}}{x_{i-j} - x_{i-j+1}}\right) f_{i-j} + \left(\frac{x - x_{i-j}}{x_{i-j+1} - x_{i-j}}\right) f_{i-j+1}. \tag{6}
$$

For the Grünwald-Letnikov operator, we simply need to index the function values over the index $k$. Thus, by substituting the indexed values of $x$ into the interpolating functions, the individual linear portions of the function take the form,

$$
g_{i-j}(k) = \frac{k - jn}{n} f_{i-j} + \frac{n - k + jn}{n} f_{i-j+1}. \tag{7}
$$

Note that the indexing is "backwards" in the variable $x$ (or time, say). That is, we have for each $j$,

$$
g_{i-j}(jn) = f_{i-j+1} \quad \text{and} \quad g_{i-j}((j+1)n) = f_{i-j}. \tag{8}
$$

Substituting the indexed function values of the piecewise linear representation into the exact formula for the Grünwald-Letnikov operator, we yield,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) \approx \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{h}{n}\right)^{(-\alpha)} \times \sum_{j=0}^{i-1} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} g_{i-j}(k). \tag{9}
$$

By rearranging, this formula can be written as,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) \approx \frac{h^{-\alpha}}{\Gamma(-\alpha)} \sum_{j=0}^{i-1} \left(\hat{C}^{(1)}_{i,j} f_{i-j} + \hat{C}^{(2)}_{i,j} f_{i-j+1}\right), \tag{10}
$$

where the constants $\hat{C}^{(1)}_{i,j}$ and $\hat{C}^{(2)}_{i,j}$ are given as,

$$
\hat{C}^{(1)}_{i,j} = \lim_{n\to\infty} n^{\alpha} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} \frac{k - jn}{n} \tag{11}
$$

$$
\hat{C}^{(2)}_{i,j} = \lim_{n\to\infty} n^{\alpha} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} \frac{n - k + jn}{n}. \tag{12}
$$

The evaluation of these summations and passage to the limit can be evaluated using well known formulae (See, e.g., [6]). This yields the expressions,

$$
\hat{C}^{(1)}_{i,0} = -\frac{\alpha}{\alpha(\alpha-1)} \tag{13}
$$

$$
\hat{C}^{(2)}_{i,0} = \frac{1}{\alpha(\alpha-1)}, \tag{14}
$$

for the limiting case of $j = 0$, as well as,

$$
\hat{C}^{(1)}_{i,j} = \frac{j^{(1-\alpha)} - (j+\alpha)(j+1)^{-\alpha}}{\alpha(\alpha-1)} \tag{15}
$$

$$
\hat{C}^{(2)}_{i,j} = \frac{(j+1)^{(1-\alpha)} - j^{(1-\alpha)} + (\alpha-1)j^{-\alpha}}{\alpha(\alpha-1)}, \tag{16}
$$

for $j = 1, \ldots, i-1$. Finally, using the observation that,

$$
\Gamma(-\alpha)\alpha(\alpha-1) = \Gamma(2-\alpha), \tag{17}
$$

we can write the approximation as,

$$
{}_0D^{\alpha}_{x_{i+1}} f(x) \approx \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} \left(C^{(1)}_{i,j} f_{i-j} + C^{(2)}_{i,j} f_{i-j+1}\right), \tag{18}
$$

where the constants are given as,

$$
C^{(1)}_{i,0} = -\alpha \tag{19}
$$

$$
C^{(2)}_{i,0} = 1, \tag{20}
$$

for $j = 0$, and,

$$C_{i,j}^{(1)} = j^{(1-\alpha)} - (j+\alpha)(j+1)^{-\alpha} \tag{21}$$

$$C_{i,j}^{(2)} = (j+1)^{(1-\alpha)} - j^{(1-\alpha)} + (\alpha-1)j^{-\alpha}, \tag{22}$$

for $j = 1, \ldots, i-1$. Note that the isolation of the limit case $j = 0$ is important for positive values of $\alpha$.
Another crucial observation is that the coefficient values themselves depend only on the index $j$, and not on the index $i$. This greatly simplifies the computation of the coefficients, as the structure of the computation defines a sum of Toeplitz matrices. Namely, if we relabel the coefficients as,

$$C_j^{(p)} = C_{i,j}^{(p)}, \tag{23}$$

for $p = 1, 2$, then we can define the matrix,

$$\mathsf{C}_0 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ -\alpha & 1 & 0 & \cdots & 0 \\ 0 & -\alpha & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\alpha & 1 \end{bmatrix}, \tag{24}$$

and the Toeplitz matrices,

$$\mathsf{C}_p = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ C_1^{(p)} & 0 & 0 & \cdots & 0 \\ C_2^{(p)} & C_1^{(p)} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C_{n-1}^{(p)} & \cdots & C_2^{(p)} & C_1^{(p)} & 0 \end{bmatrix}, \tag{25}$$

for $p = 1, 2$. The matrix discretization of the Grünwald-Letnikov operator is then given by the expression,

$$\mathsf{S}^{(\alpha)} = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \left(\mathsf{C}_0 + \mathsf{C}_1 + \mathsf{C}_2\mathsf{J}\right), \tag{26}$$

where the matrix $\mathsf{J}$ is a circulant shift operator, defined as,

$$\mathsf{J} = \left[\boldsymbol{e}_n \; \boldsymbol{e}_1 \; \boldsymbol{e}_2 \cdots \boldsymbol{e}_{n-1}\right]. \tag{27}$$

This Toeplitz structure thereby leads to greatly simplified MATLAB code for generating the matrix discretization of the operator, as can be seen in Sect. 6. Specifically, since there are $2(n-1)$ coefficients that require computing, they can be computed in order $\mathcal{O}(n)$ time.

## 3 Arbitrary spacing

For arbitrary spacing, we take the same approach, however, the formulae are slightly more complicated. In order to calculate the value of the Grünwald-Letnikov operator at the point, $x_{i+1}$, we divide the interval $[x_1, x_{i+1}]$ into $n$ points. In turn, we divide each individual interval $[x_{i-j}, x_{i-j+1}]$ into $n_{j+1}$ points. This partitioning and indexing of intervals is shown in Fig. 1.

We further denote the partial sums of the divisions of each interval as,

$$N_j = \sum_{k=1}^{j} n_k, \tag{28}$$

such that $N_0 = 0$, $N_1 = n_1$, $N_2 = n_1 + n_2$, and so on until $N_i = n$. In this case, the indexing of the function values over the irregularly spaced intervals takes the form,

$$g_{i-j}(k) = \frac{k - N_j}{n_{j+1}} f_{i-j} + \frac{N_{j+1} - k}{n_{j+1}} f_{i-j+1}. \tag{29}$$

For this partitioning of the intervals, the formula for the Grünwald-Letnikov operator with lower limit $x_1$ now reads,

$$\begin{aligned} {}_{x_1}D_{x_{i+1}}^{\alpha} f(x) \approx \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{x_{i+1} - x_1}{n}\right)^{(-\alpha)} \\ \times \sum_{j=0}^{i-1} \sum_{k=N_j}^{N_{j+1}-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} g_{i-j}(k) \end{aligned} \tag{30}$$

In a similar manner to the evenly spaced case, this can be written as,

$$ {}_{x_1}D_{x_{i+1}}^{\alpha} f(x) \approx \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} \left(K_{i,j}^{(1)} f_{i-j} + K_{i,j}^{(2)} f_{i-j+1}\right) \tag{31}$$

where the coefficients are computed by evaluating the sums over $k$ analytically, followed by the passage to the limit. The key observation necessary in evaluating the limits is the fact that for the values of $n_j$ and $N_j$, we have the asymptotic relations,

$$\frac{n_{j+1}}{n} \propto \frac{x_{i-j+1} - x_{i-j}}{x_{i+1} - x_1} \tag{32}$$

$$\frac{N_j}{n} \propto \frac{x_{i+1} - x_{i-j+1}}{x_{i+1} - x_1} \tag{33}$$

$$\frac{N_{j+1}}{n} \propto \frac{x_{i+1} - x_{i-j}}{x_{i+1} - x_1}. \tag{34}$$

In this vein, if we define the differences,

$$h_{i-j} = x_{i-j+1} - x_{i-j} \tag{35}$$

$$u_{i,j} = x_{i+1} - x_{i-j+1} \tag{36}$$

$$v_{i,j} = x_{i+1} - x_{i-j}, \tag{37}$$

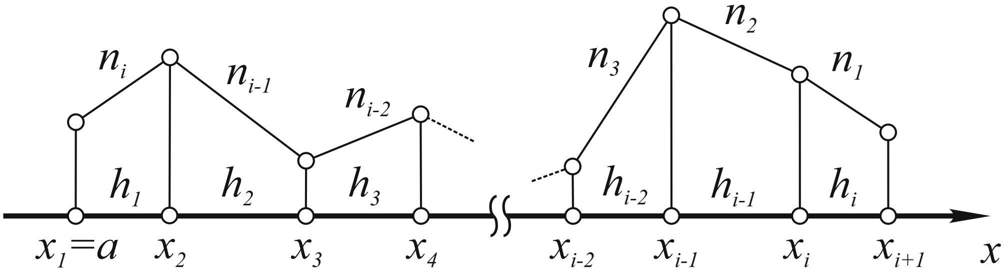

**Fig. 1** Partitioning and indexing of the intervals. Each individual interval $[x_{i-j}, x_{i-j+1}]$ is divided into $n_{j+1}$ points, where the ratio of $n_{j+1}$ to $n$ is asymptotically equal to the ratio of the spacing $h_{i-j}$ to the total length $x_{i+1} - x_1$

then the coefficients of the approximation are given as,

$$K^{(1)}_{i,0} = -\alpha h_i^{-\alpha} \tag{38}$$

$$K^{(2)}_{i,0} = h_i^{-\alpha}, \tag{39}$$

for the case $j = 0$, and,

$$K^{(1)}_{i,j} = \frac{(u^{1-\alpha}_{i,j} - (u_{i,j} + \alpha h_{i-j})v^{-\alpha}_{i,j})}{h_{i-j}} \tag{40}$$

$$K^{(2)}_{i,j} = \frac{(v^{1-\alpha}_{i,j} - (v_{i,j} - \alpha h_{i-j})u^{-\alpha}_{i,j})}{h_{i-j}}, \tag{41}$$

for $j = 1, \ldots, n-1$. In the case of arbitrary spacing, the coefficients do not have a Toeplitz structure, and the computation of the general coefficients is therefore an order $\mathcal{O}(n \log n)$ operation.

## 4 Initial value and interpolation over the first interval

An issue that is of computational importance is the value of the Grünwald-Letnikov operator at the lower limit, $x = x_1$. Depending on the value of $f(x_1)$ and the order of the operator $\alpha$, the initial value,

$$f^{(\alpha)}(x_1) = \lim_{x=x_1} {}_{x_1}D^{\alpha}_{x} f(x), \tag{42}$$

can take on any value in $[-\infty, \infty]$. Furthermore, the function resulting from the Grünwald-Letnikov operator can have strong asymptotic behavior over this initial interval. The interpolation function can be derived in a straightforward manner from the interpolation function over the first interval. In this case, the Newton form of the interpolating function is more appropriate, namely,

$$g_1(x) = f_1 + \frac{f_2 - f_1}{h_1}(x - x_1), \tag{43}$$

where $h_1 = x_2 - x_1$. By applying the monomial rule for the Grünwald-Letnikov operator, we yield the function,

$${}_{x_1}D^{\alpha}_{x} g_1(x) = \frac{(x - x_1)^{-\alpha}}{\Gamma(1-\alpha)} f_1 + \frac{(x - x_1)^{1-\alpha}}{h_1 \Gamma(2-\alpha)}(f_2 - f_1). \tag{44}$$

This function can be used to interpolate the result over the interval $[x_1, x_2]$, as well as determine the value at $x = x_1$ by means of a limiting process.

## 5 Error analysis

As with any numerical approximation, an essential component of the approximation is its error analysis. For the trapezoidal rule proposed herein, an error term can be estimated based on the remainder formula for Lagrange interpolation. This derivation is followed by the error analysis for arbitrary spacing.

### 5.1 Evenly spaced points

We can write an exact representation of a function, $f(x)$ over the interval $[x_{i-j}, x_{i-j+1}]$, by means of the Lagrange interpolation formula and its error term. Specifically, a continuous function $f(x)$ can be written as,

$$f(x) = g(x) + r(x), \tag{45}$$

where the function $g(x)$ is again the piecewise linear representation of $f(x)$ and $r(x)$ is a remainder function. The remainder is also defined piecewise as,

$$r(x) = r_{i-j}(x) \qquad \text{for} \qquad x \in [x_{i-j}, x_{i-j+1}]. \tag{46}$$

The functions, $r_{i-j}(x)$, are given by the Lagrange remainder,

$$r_{i-j}(x) = \frac{1}{2} f''(\xi_{i-j}(x))(x - x_{i-j})(x - x_{i-j+1}), \tag{47}$$

where $\xi_{i-j}(x) \in [x_{i-j}, x_{i-j+1}]$ is an unknown value that depends on $x$. Thus, by applying the Grünwald-

Letnikov operator to both sides of Eq. (45), we obtain,

$$ {}_0D^{\alpha}_{x_{i+1}} f(x) = {}_0D^{\alpha}_{x_{i+1}} g(x) + {}_0D^{\alpha}_{x_{i+1}} r(x). \tag{48}$$

The remainder term can be computed by substituting the remainder function into the formula for the Grünwald-Letnikov operator obtained in Eq. (4), namely,

$$\begin{aligned} {}_0D^{\alpha}_{x_{i+1}} r(x) &= \lim_{n\to\infty} \frac{1}{\Gamma(-\alpha)} \left(\frac{h}{n}\right)^{(-\alpha)} \\ &\quad \times \sum_{j=0}^{i-1} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} r_{i-j}\left(x_{i+1} - k\frac{x_{i+1}}{in}\right). \end{aligned} \tag{49}$$

By substituting the indexed values of $x$ into the remainder function, we have,

$$ r_{i-j}(k) = \frac{1}{2} f''(\xi_{i-j}(k)) \frac{h^2}{n^2} (jn-k)((j+1)n-k). \tag{50}$$

Substituting this relation into the previous equation yields,

$$\begin{aligned} {}_0D^{\alpha}_{x_{i+1}} r(x) &= \lim_{n\to\infty} \frac{1}{2\Gamma(-\alpha)} \left(\frac{h}{n}\right)^{(2-\alpha)} \\ &\quad \times \sum_{j=0}^{i-1} \sum_{k=jn}^{(j+1)n-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f''(\xi_{i-j}(k))(jn-k)((j+1)n-k). \end{aligned} \tag{51}$$

This relation can be written more compactly by defining the constants,

$$ p_k = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}, \tag{52}$$

and

$$ q_k = (jn-k)((j+1)n-k). \tag{53}$$

The expression for the Grünwald-Letnikov operator applied to the remainder term can thereby be written as,

$$\begin{aligned} &{}_0D^{\alpha}_{x_{i+1}} r(x) \\ &\quad = \lim_{n\to\infty} \frac{1}{2} \left(\frac{h}{n}\right)^{(2-\alpha)} \sum_{j=0}^{i-1} \sum_{k=jn}^{(j+1)n-1} p_k q_k f''(\xi_{i-j}(k)). \end{aligned} \tag{54}$$

The coefficients $p_k$ in the summation are given by a recurrence relation [7] as,

$$ p_0 = 1 \tag{55}$$

$$ p_k = \left(1 - \frac{\alpha+1}{k}\right) p_{k-1} \qquad \text{for} \qquad k = 1, 2, \ldots \tag{56}$$

Hence, if $\alpha < 0$, then the coefficients, $p_k$, are zero or greater for all $k$. Similarly, if $0 \leq \alpha \leq 1$, then clearly, $p_0 > 0$, but $p_k < 0$ for $k > 0$. In a similar manner, the coefficients $q_k$ are determined by a polynomial function which is quadratic in $k$, and has roots at $k = jn$ and $k = (j+1)n$. Between these roots, the function is negative, and hence we can say that the coefficients $q_k \leq 0$ for all $k$. Putting this together, we can establish the fact that the product of the coefficients does not change sign when $\alpha \leq 1$, i.e.,

$$ p_k q_k \leq 0 \quad \forall k \quad \text{for} \quad \alpha \leq 0 \tag{57}$$

$$ p_k q_k \geq 0 \quad \forall k \quad \text{for} \quad 0 \leq \alpha \leq 1. \tag{58}$$

The product $p_k q_k$ represents successive values of a continuous function, which does not change sign over the interval $[x_{i-j}, x_{i-j+1}]$. Therefore, if the second derivative $f''(x)$ is continuous over the interval, then by the generalized mean value theorem [15], we can write,

$$\begin{aligned} &{}_0D^{\alpha}_{x_{i+1}} r(x) \\ &\quad = \lim_{n\to\infty} \frac{1}{2} \left(\frac{h}{n}\right)^{(2-\alpha)} \sum_{j=0}^{i-1} f''(\eta_{i-j}) \sum_{k=jn}^{(j+1)n-1} p_k q_k, \end{aligned} \tag{59}$$

where $\eta_{i-j}$ is a value in the interval $[x_{i-j}, x_{i-j+1}]$. By rearranging, we have,

$$\begin{aligned} &{}_0D^{\alpha}_{x_{i+1}} r(x) \\ &\quad = \frac{h^{2-\alpha}}{2} \sum_{j=0}^{i-1} f''(\eta_{i-j}) \left( \lim_{n\to\infty} n^{\alpha-2} \sum_{k=jn}^{(j+1)n-1} p_k q_k \right). \end{aligned} \tag{60}$$

Finally, by evaluating the sum and the limit, we have,

$$ {}_0D^{\alpha}_{x_{i+1}} r(x) = \frac{h^{2-\alpha}}{2\Gamma(3-\alpha)} \sum_{j=0}^{i-1} \kappa_j f''(\eta_{i-j}), \tag{61}$$

where the coefficients, $\kappa_j$, are given as,

$$ \kappa_0 = \alpha, \tag{62}$$

and

$$ \kappa_j = (j+1)^{1-\alpha}(2j+\alpha) - j^{1-\alpha}(2(j+1)-\alpha), \tag{63}$$

for $j = 1, 2, \ldots, i-1$. Since these coefficients depend only on the index $j$, they again define a Toeplitz matrix. That is, for the purpose of computing error bounds, we can define the matrix,

$$\mathsf{R} = \frac{h^{2-\alpha}}{2\Gamma(3-\alpha)} \begin{bmatrix} \kappa_0 & 0 & 0 & \cdots & 0 \\ \kappa_1 & \kappa_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \kappa_{n-3} & \cdots & \kappa_1 & \kappa_0 & 0 \\ \kappa_{n-2} & \cdots & \kappa_2 & \kappa_1 & \kappa_0 \end{bmatrix}. \tag{64}$$

This matrix can be used to compute upper and lower bounds on the residual error of the computed result. Provided the second derivative can be bounded, this can be done regardless if the exact solution is known or not. In order to compute the error bounds, we substitute the result in Eq. (61) into (48) to yield,

$$ {}_0D^{\alpha}_{x_{i+1}} f(x) - {}_0D^{\alpha}_{x_{i+1}} g(x) = \frac{h^{2-\alpha}}{2\Gamma(3-\alpha)} \sum_{j=0}^{i-1} \kappa_j f''(\eta_{i-j}). \tag{65}$$

This expression describes the residual difference between the exact solution and the trapezoidal rule; the relation itself is exact, recalling however that the values $\eta_{i-j}$ are generally unknown.

By defining the vectors,

$$\boldsymbol{f}^{(\alpha)} = \begin{bmatrix} {}_0D^{\alpha}_{x_2} f(x) \\ {}_0D^{\alpha}_{x_3} f(x) \\ \vdots \\ {}_0D^{\alpha}_{x_n} f(x) \end{bmatrix} \quad \text{and} \quad \boldsymbol{g}^{(\alpha)} = \begin{bmatrix} {}_0D^{\alpha}_{x_2} g(x) \\ {}_0D^{\alpha}_{x_3} g(x) \\ \vdots \\ {}_0D^{\alpha}_{x_n} g(x) \end{bmatrix}, \tag{66}$$

as well as,

$$\boldsymbol{f}''_{\eta} = \begin{bmatrix} f''(\eta_1) \\ f''(\eta_2) \\ \vdots \\ f''(\eta_{n-1}) \end{bmatrix}, \tag{67}$$

the relation can be written for all points, $x_2, \ldots, x_n$ as,

$$\boldsymbol{f}^{(\alpha)} - \boldsymbol{g}^{(\alpha)} = \mathsf{R}\boldsymbol{f}''_{\eta}. \tag{68}$$

This relation can be used to bound the residual error of the computation. Specifically, if the second derivative is continuous, then by the extreme value theorem [15], it can be bounded over each interval such that,

$$b_k \leq f''(x) \leq B_k, \quad \text{for} \quad x \in [x_k, x_{k+1}]. \tag{69}$$

We therefore define the vectors,

$$\boldsymbol{b}_{\mathrm{L}} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{b}_{\mathrm{U}} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n-1} \end{bmatrix}. \tag{70}$$

To determine a bound, the sign of the coefficients in the matrix $\mathsf{R}$ are important. Firstly, for the range of $\alpha$ that is of interest, i.e., $\alpha < 1$, the common factor of the matrix $\mathsf{R}$ is always positive such that,

$$\frac{h^{2-\alpha}}{2\Gamma(3-\alpha)} \geq 0, \quad \text{for} \quad \alpha < 1, h > 0. \tag{71}$$

Further, since for $\alpha < 0$ the coefficients $\kappa_k$ were obtained as the sum of terms that are less than or equal to zero, the coefficients are negative, i.e.,

$$\kappa_j < 0 \quad \text{for} \quad j = 0, \ldots, n-2, \quad \text{and} \quad \alpha < 0. \tag{72}$$

Similarly, for the range $0 < \alpha < 1$ the coefficients were obtained as the sum of terms that are greater or equal to zero, hence the coefficients are positive, i.e.,

$$\kappa_j > 0 \quad \text{for} \quad j = 0, \ldots, n-2, \quad \text{and} \quad 0 < \alpha < 1. \tag{73}$$

We may now take advantage of the fact that in the specified cases, namely when $\alpha < 1$, the non-zero entries of the matrix $\mathsf{R}$ all have the same sign. When the Grünwald-Letnikov operator is computed with $0 \leq \alpha \leq 1$, then the operator represents fractional differentiation; the non-zero entries of $\mathsf{R}$ are all positive. The residual error is therefore bounded by the relation,

$$\mathsf{R}\boldsymbol{b}_{\mathrm{L}} \leq \boldsymbol{f}^{(\alpha)} - \boldsymbol{g}^{(\alpha)} \leq \mathsf{R}\boldsymbol{b}_{\mathrm{U}} \quad \text{for} \quad 0 \leq \alpha \leq 1. \tag{74}$$

Similarly, for the case $\alpha < 0$, the Grünwald-Letnikov operator represents fractional integration; the non-zero entries of $\mathsf{R}$ are all negative, and hence the residual error is bounded by,

$$\mathsf{R}\boldsymbol{b}_{\mathrm{U}} \leq \boldsymbol{f}^{(\alpha)} - \boldsymbol{g}^{(\alpha)} \leq \mathsf{R}\boldsymbol{b}_{\mathrm{L}} \quad \text{for} \quad \alpha \leq 0. \tag{75}$$

With these relations it is therefore possible to bound the residual error for all cases $\alpha < 1$, if a bound on the second derivative can be computed. For a given function, if its second derivative can be computed in a straightforward manner, then the bounds over each interval can be computed, e.g., using interval analysis [16]. The obvious benefit to this bound is that a simple algorithm can be derived to adjust the spacing such that the residual is guaranteed to be smaller than a given tolerance. For such an algorithm to be efficient, we require the error bound for arbitrary spacing, which is developed in the following.

### 5.2 Arbitrary spacing

The error bound for arbitrary spacing is computed in a similar manner as to that for even spacing. The remainder function, discretized for arbitrary spacing is given as,

$$r_{i-j}(k) = \frac{1}{2} f''(\xi_{i-j}(k)) \frac{(x_{i-j+1}-x_{i-j})^2}{n_{i+1}^2}(N_{j+1}-k)(N_j-k), \tag{76}$$

over the interval $[x_{i-j}, x_{i-j+1}]$, where again, $\xi_{i-j}(k) \in [x_{i-j}, x_{i-j+1}]$, is an unknown value. As with the evenly spaced case, the Grünwald-Letnikov operator applied to the error term takes the form,

$$_{x_1}D^{\alpha}_{x_{i+1}} r(x) = \frac{1}{2\Gamma(3-\alpha)} \sum_{j=0}^{n-2} \kappa_{i,j} f''(\eta_{i-j}), \tag{77}$$

with $\eta_{i-j} \in [x_{i-j}, x_{i-j+1}]$. The coefficients for the error bound are given as,

$$\kappa_{i,0} = \alpha h_i^{2-\alpha}, \tag{78}$$

for $j = 0$, as well as,

$$\kappa_{i,j} = (v_{i,j}^{2-\alpha} - u_{i,j}^{2-\alpha})\alpha + (u_{i,j}^{1-\alpha} v_{i,j} - u_{i,j} v_{i,j}^{1-\alpha})(\alpha - 2) \tag{79}$$

for $j = 1, \ldots, i-1$, with $u_{i,j}$ and $v_{i,j}$ given as in Eqs. (36) and (37). The identical argument follows in terms of the signs of the coefficients, and hence the error bounds are given in terms of Eqs. (74) and (75).

### 5.3 Backwards compatibility

It is straightforward to show that the formulae are backwards compatible with the known error formulae for the trapezoidal rule. For the case $\alpha = 1$, the trapezoidal rule is equivalent to the backward-difference formula. In this case, simple substitution shows that $\kappa_0 = 1$, and $\kappa_j = 0$ for $j = 1, \ldots, n-2$. Therefore, the error bound is given in terms of the matrix,

$$\mathsf{R} = \frac{h}{2}\mathsf{I}, \tag{80}$$

which is the known formula for the error of backward-differences [17]. For the case of $\alpha = -1$, we have simple integration. Again, simple substitution shows that, $\kappa_j = -1$ for $j = 0, 1, \ldots, n-2$, and therefore the bound matrix is given as,

$$\mathsf{R} = -\frac{h^3}{12}\mathsf{L}, \tag{81}$$

where $\mathsf{L}$ is the $(n-1) \times (n-1)$ lower triangular matrix,

$$\mathsf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{82}$$

Again, this is the well known error term for the classical trapezoidal rule for integration [17].

## 6 Numerical implementation and testing

### 6.1 MATLAB Code

The following MATLAB code[1] generates a matrix $\mathsf{S}$ for the trapezoidal approximation to the Grünwald-Letnikov operator for evenly spaced points. The input to the function is the spacing, `h`, the number of abscissae `n`, and the order, `nu`:

```
j = 1:n-2;
u = j.^(1-nu) - (j+nu).*(j+1).^(-nu);
v = (j + 1).^(1-nu) ...
    - j.^(1-nu)+(nu-1) * j.^(-nu);
S = diag([0,ones(1,n-1)]) ...
    + diag(-nu*ones(1,n-1),-1);
S = S + toeplitz([0,0,u],zeros(n,1));
S = S + ...
    circshift(toeplitz([0,0,v], ...
    zeros(n,1)),[0,1]);
S = h^(-nu)*S/gamma(2-nu);
```

Note that this code has been arranged for compactness and readability; it is not necessarily optimized for speed or memory allotment. The code for generating the error bound matrix is given as:

```
R = nu * eye(n-1) ;
u = ( j+1 ).^(1-nu) .* ( 2*j + nu ) ...
   - j.^(1-nu) .* ( 2*(j+1) - nu ) ;
R = R + toeplitz([0,u],zeros(n-1,1)) ;
R = ( h^(2-nu) )*R / ( 2*gamma(3-nu) ) ;
```

As for the code for arbitrary spacing, the structure of the coefficients is much the same as that for evenly spaced points, except for the notion of the Toeplitz matrix. The necessary matrix can be computed by means of a double nested loop, with $i$ ranging from 1 to $n-1$, and $j$ ranging from 1 to $i-1$. The complexity of computing all coefficients for arbitrary spacing is therefore $\mathcal{O}(n \log n)$.

### 6.2 Numerical testing

In this section we utilize four numerical tests to demonstrate the functionality of the newly proposed formulae:

[1] The following code was written and tested in MATLAB R2013b; no special toolboxes are required. Note that `nu` is used instead of `alpha` to aid readability.

1. Testing the formulae for a linear function with even spacing, showing that the results are exact. For this test, we compare the results to the classic Grünwald-Letnikov approximation and show that it is not exact (even for linear functions) and only achieves accurate results for a very large number of abscissae.
2. Testing the formulae for a linear function with arbitrary spacing, in this case with randomized abscissae. The results of the new formulae are also exact in this case. This test also demonstrates the interpolation formula for the first interval derived in Sect. 4.
3. We then demonstrate the full functionality of the formulae by applying the numerical Grünwald-Letnikov operator to a sinusoidal function. We compare the classic approximation to the new trapezoidal rule with both evenly spaced points, as well as arbitrary spacing. For the case of arbitrary spacing, the nodes were chosen such that they yield an optimal piecewise-linear approximation of the sinusoid.
4. Finally, we use the new error bound to develop a simple algorithm for calculating the Grünwald-Letnikov integral or derivative of a function to a given accuracy tolerance without knowing the exact solution, but knowing only the second derivative.

The results of the first test are shown in Fig. 2 for the values $\alpha = \frac{3}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{3}{2}$. As expected, applying the new formula to a linear function evaluated at evenly spaced points yields exact results. For this particular test we chose $n = 31$ points, such that the spacing[2] was $h = \frac{1}{10}$. Even for this relatively small spacing the classical approximation yields a significant error in the approximation. For a much larger number of abscissae the classic approximation will become more accurate; however, due to the fact that we are always restricted to finite precision arithmetic, the classic approximation will always have a bias.

The second experiment is simply to show that the new approximation also yields exact results when applied to a linear function evaluated at arbitrarily spaced points. The results are shown in Fig. 3, whereby the function was evaluated at $n = 10$ arbitrarily spaced points, and the same values of $\alpha$ as above. This test shows that the results are still exact for linear functions evaluated at large intervals of unequal lengths, and that the interpolation formula over the interval $[x_1, x_2]$ is also exact for this case.

To demonstrate the possibilities for applying the new formulae in practise, they have been applied with $\alpha = -\frac{1}{2}$ to the trigonometric function, $f(x) = \cos x$; the results are shown in Fig. 4. Firstly, the classic approximation was applied to

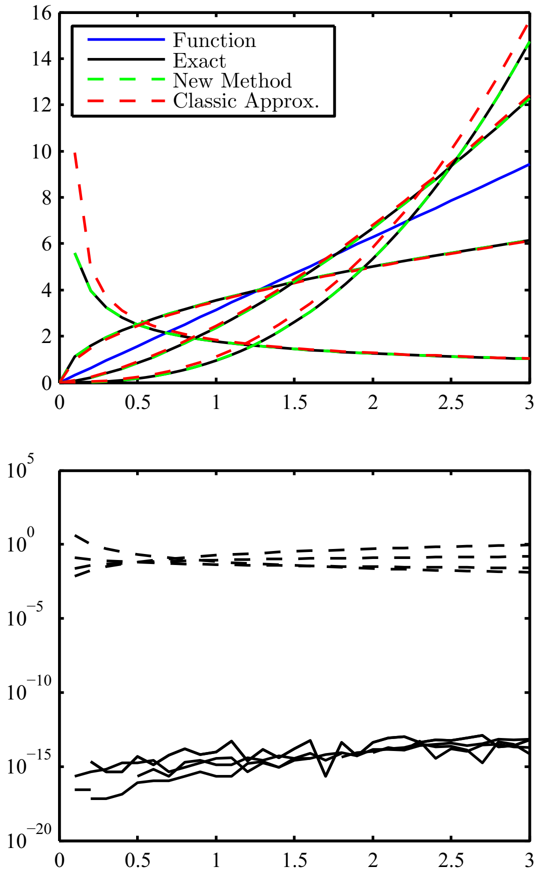

**Fig. 2** (*top*) A linear function and the exact and approximate Grünwald-Letnikov operator applied to it with $\alpha = \frac{3}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{3}{2}$. (*bottom*) Residual errors for the two approximations (*dashed line*) Classical (*line*) New approximation. The new formula is exact to the limits of double precision arithmetic

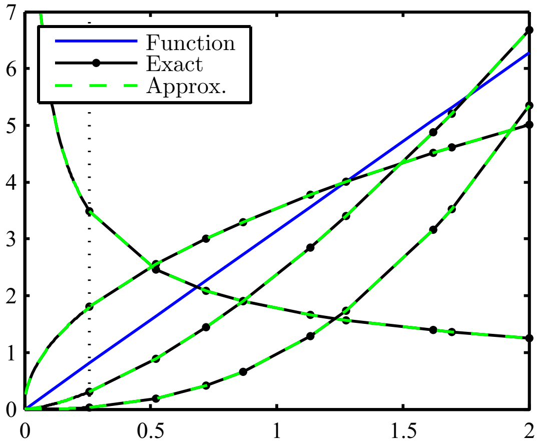

**Fig. 3** A linear function and the exact and approximate Grünwald-Letnikov operator applied to it with $\alpha = \frac{3}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{3}{2}$. The results of the new approximation are superimposed on the exact solution. The function was evaluated at $n = 10$ arbitrarily spaced points, showing that the approximation is still exact for large, uneven spacing. The interpolation formula has been used over the first interval, demarcated by the *vertical dotted line*

[2] The choice of $h = 0.1$ is not the best since it has no exact binary representation; it was chosen to demonstrate that the formulas remain accurate also in this case.

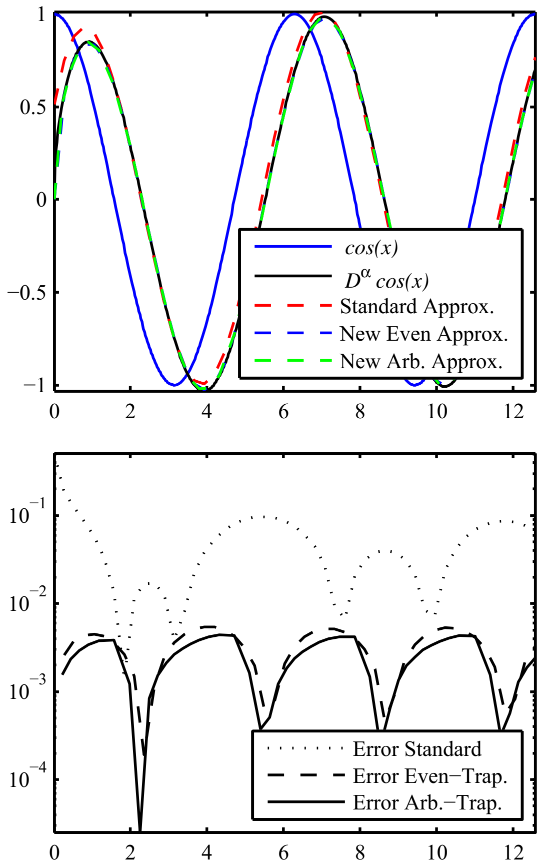

**Fig. 4** (*top*) The function $f(x) = \cos x$ and the exact and approximated Grünwald-Letnikov operators applied to it with $\alpha = -\frac{1}{2}$. Shown are the classic approximation for $n = 49$ evenly spaced points, the new approximation for $n = 49$ evenly spaced points, as well as arbitrary spacing to provide a better piecewise linear approximation to $f(x)$. (*bottom*) Residual errors for the three approximations. The new approximations are about two orders of magnitude better than the classic approximation for the same number of points

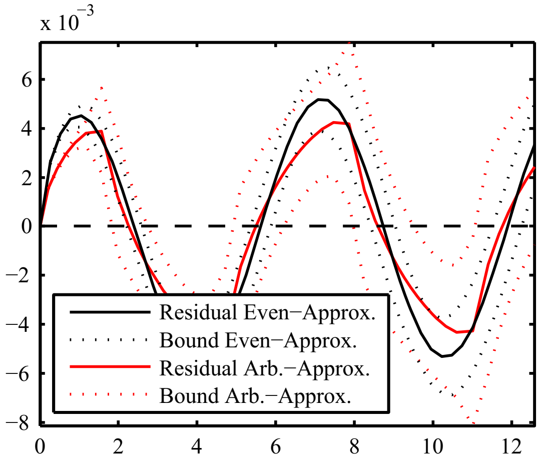

**Fig. 5** Residuals and analytic bounds for the example $f(x) = \cos x$. Using a better choice of spacing can reduce the residual error

the function with $n = 49$ evenly spaced points. Secondly, the trapezoidal rule for even-spacing was applied to the same points. Thirdly, we determined a spacing numerically which best represents the cosine function as a piecewise linear function, with the same number of nodes. Clearly the results show that the classic approximation to the Grünwald-Letnikov operator still has significant bias, especially near the initial values. The evenly spaced trapezoidal rule provides a much better approximation. In fact, in the residual plot it is shown to be roughly one to two orders-of-magnitude more accurate. The residual plot also shows that a better choice of spacing, in combination with the trapezoidal rule for arbitrary spacing, can be used to yield more accurate results with the same number of points. Figure 5 shows the residuals for the two new methods for this example. In addition, it shows the computed bounds on the residuals for both methods. Clearly, corroborating the theory, the residual lies between the two computed bounds. An interesting effect, is that although the arbitrary spacing has made the solution more accurate, the bound about the residual is slightly larger. This is due to the fact that in some places where the second derivative is small, the spacing, $h$, was chosen to be larger. In this case, the error bound increases with $h^{\frac{5}{2}}$, and therefore increases with increased spacing values.

As a final example, the error bound developed in this paper can be used to compute a numerical approximation to the Grünwald-Letnikov operator applied to a continuous function to within a given tolerance of accuracy. As an example, we chose the function,

$$f(x) = \mathrm{e}^{-\frac{x}{3}} \cos\left(x^{\frac{9}{5}}\right), \tag{83}$$

since its second derivative can be computed with relative ease, but its fractional derivative not. The aim was to compute the half-derivative, $\alpha = \frac{1}{2}$, at a set of points in the interval $x \in [0, (\frac{13\pi}{2})^{\frac{5}{9}}]$, to within a tolerance of $t = \pm 0.025$. Choosing an initial partition of $n = 15$ evenly spaced points, we computed the bounds on the error—which only involves bounding the second derivative. For any point where the residual exceeded the tolerance, we added the mid-point in the preceding interval to the set of abscissae. This process is continued until the residual bound for all points lies within the tolerance $t = \pm 0.025$. Finally, the trapezoidal rule is computed at the set of adapted abscissae, which is guaranteed to be accurate to within the tolerance. The results of this process are shown in Fig. 6. Here, the initial trapezoidal approximation with $n = 15$ is shown. Further, the final set of function evaluations are shown, following the adaptive computation. The result is shown for the initial trapezoidal approximation, as well as the final adaptive computation of the half-derivative. In addition, the solution is also interpolated over the first interval; since it is a half-derivative with

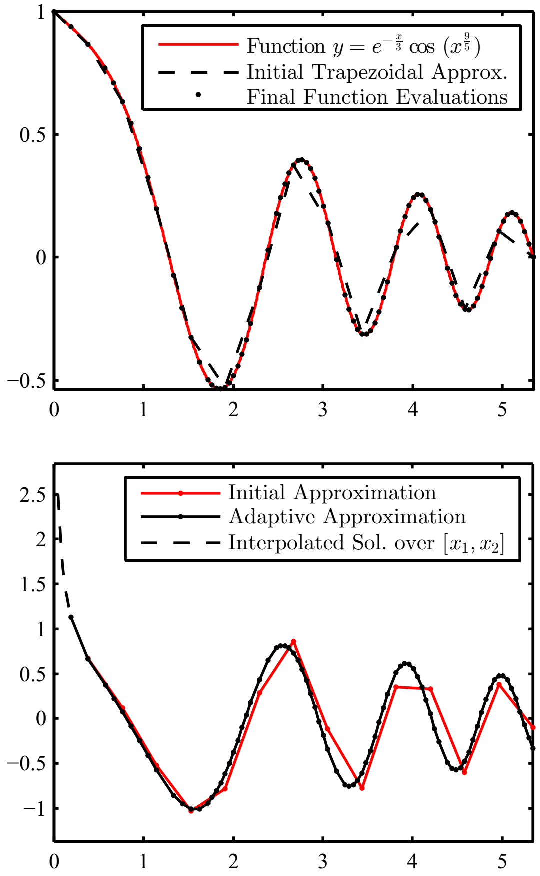

**Fig. 6** The adaptive computation for determining the abscissae such that the residual error is guaranteed to be below a set tolerance. (*top*) The test function $f(x)$ with its initial and final piecewise approximations. (*bottom*) The initial approximation of the half derivative as well as the final approximation computed to within $t = \pm 0.025$

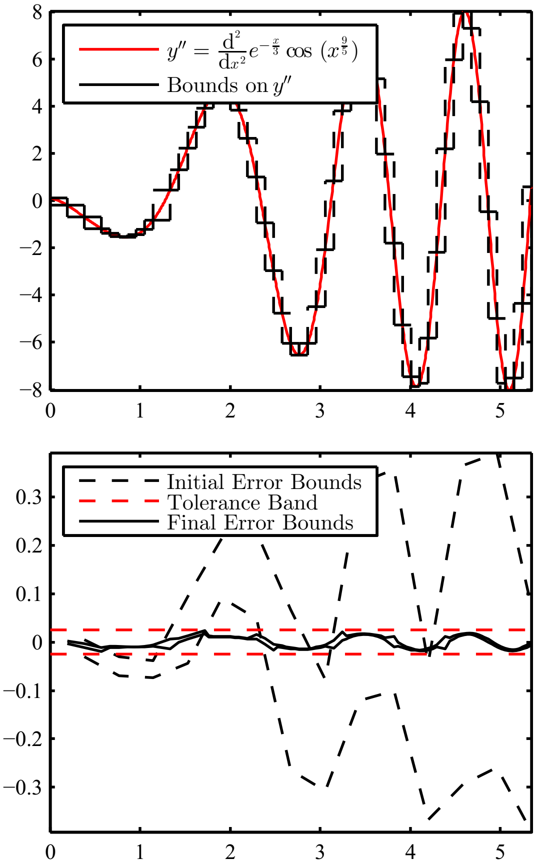

**Fig. 7** Final bounds on the second derivative for the adaptive scheme, as well as the computed residual bounds for the initial and final sets of abscissae. Clearly, the final error bounds are within the tolerance, and hence the half-derivative is computed to within $t = \pm 0.025$ without knowledge of the exact solution

$f(x_1) = 1$, it tends asymptotically to $+\infty$. Figure 7 shows the bounds on the second derivative of the function at the final partitioning, as well as the residual bounds. The initial trapezoidal approximation with $n = 15$ is shown to have large bounds, which are outside of the tolerance. Following the adaptive scheme, the final residual bounds lie entirely within the proposed tolerance.

## 7 Outlook and conclusion

In this paper, we derived formulae for the exact application of the Grünwald-Letnikov operator to piecewise linear functions, essentially a trapezoidal rule for the Grünwald-Letnikov operator. Since the formulae were derived for arbitrary spacing, the formulae can be used to advantage by selecting abscissae such that higher accuracy is obtained with fewer points. Further, we derived an analytic bound on the residual error for the computation. The error bound can be used to compute the result to within a given tolerance. The error bound can also be used to compute a rigorous error estimate when applying the short memory principle in practical computations. Future work is to extend the result to higher order formulas, which should be possible based on the results of the present paper.

**Acknowledgements** Open access funding provided by Montanuniversität Leoben.

## References

1. Harker M, O'Leary P (2014) Numerical solution of fractional order differential equations via matrix-based methods, Fractional Calculus: theory, vol 1. Nova Science Publishers, Hauppauge, NY
2. Podlubny I (2000) Matrix approach to discrete fractional calculus. Fract Calc Appl Anal 3(4):359–386

3. Harker M, O'Leary P (2015) Sylvester equations and the numerical solution of partial fractional differential equations. J Comput Phys 293:370–384
4. Podlubny I, Chechkin A, Skovranek T, Chen YQ, Jara BMV (2009) Matrix approach to discrete fractional calculus II: partial fractional differential equations. J Comput Phys 228(8):3137–3153
5. Grünwald A (1867) Ueber "begrenzte" derivationen und deren anwendung. Zeitschrift f Mathematik u Physik 12(6):441–480
6. Miller K, Ross B (1993) An introduction to the fractional calculus and fractional differential equations. Wiley, New York
7. Podlubny I (1999) Fractional differential equations, mathematics in science and engineering, vol 198. Academic Press, San Diego
8. Diethelm K, Ford N, Freed A, Luchko Y (2005) Algorithms for the fractional calculus: a selection of numerical methods. Comput Methods Appl Mech Eng 194:743–773
9. Harker M, O'Leary P (2014) Polynomial accurate numerical fractional order integration and differentiation. In: International conference on fractional differentiation and its applications (ICFDA), 2014, pp 1–6
10. Oldham K, Spanier J (1974) The fractional calculus: theory and applications of differentiation and integration to arbitrary order. Dover Publications Inc, Minneola, USA
11. TenreiroMachado J (2015) Numerical calculation of the left and right fractional derivatives. J Comput Phys 293:96–103
12. Podlubny I, Skovranek T, Jara BMV, Petras I, Verbitsky V, Chen Y (2013) Matrix approach to discrete fractional calculus III: non-equidistant grids, variable step length and distributed orders. Philos Trans R Soc Lond A Math Phys Eng Sci 371(1990)
13. Press W, Teukolsky S, Vetterling W, Flannery B (2007) Numerical recipes: the art of scientific computing, 3rd edn. Cambridge University Press, Cambridge
14. Golub G, Van Loan C (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore
15. Courant R, John F (1989) Introduction to calculus and analysis, classics in mathematics, vol I & II. Springer, Berlin
16. Moore R, Kearfott R, Cloud M (2009) Introduction to interval analysis. SIAM, Philadelphia
17. Burden R, Faires J (2005) Numerical Analysis, 8th edn. Thomson Learning, Inc, Belmont