



A unified Explicit form for difference formulas for fractional and classical derivatives and applications

Wickramaarachchilage Anura Gunarathna¹, Haniffa Mohamed Nasir^{2,*}, and Wasantha Bandara Daundasekera³

¹Department of Physical Sciences, Faculty of Applied Science, Rajarata University, Sri Lanka.

²FracDiff Research Group, Department of Mathematics, Sultan Qaboos University, Al-Khod 123, Muscat, Sultanate of Oman..

³Department of Mathematics, University of Peradeniya, Sri Lanka.

Abstract

A unified explicit form for difference formulas to approximate fractional and classical derivatives is presented. The formula gives finite difference approximations for any classical derivative with a desired order of accuracy at any nodal point in computational domain. It also gives Grünwald type approximations for fractional derivatives with arbitrary order of approximation at any nodal point. Thus, this explicit form unifies approximations of both types of derivatives. Moreover, for classical derivatives, it also provides various finite difference formulas such as forward, backward, central, staggered, compact, non-compact, etc. Efficient computations of the coefficients of the difference formulas are also presented leading to automating the solution process of differential equations with a given higher order accuracy. Some basic applications are presented to demonstrate the usefulness of this unified formulation.

Keywords. Fractional derivative, Shifted Grünwald approximation, Lubich Generators, Compact finite difference formula, Boundary value problem.
2010 Mathematics Subject Classification. 65L05, 34K06, 34K28.

1. INTRODUCTION

Fractional calculus has a history that goes back to L'Hospital, Leibniz, and Euler [5, 15]. A historical account of early works on fractional calculus can be found, for eg., in [25]. Fractional integrals and fractional derivatives are extensions of the integer-order (or we call classical) integrals and derivatives to a real or complex order. Various definitions of fractional derivatives have been proposed in the past, among which the Riemann-Liouville, Grünwald-Letnikov and Caputo derivatives are common and established.

Recently, fractional calculus found its way into the application domain in science and engineering [2, 10, 18, 20, 28]. Fractional derivatives are also found to be more suitable to describe anomalous transport in an external field derived from the continuous time random walk [3], resulting in a fractional diffusion equation (FDE). The FDE involves fractional derivatives either in time, in space or in both variables.

Difference approximation in the form of an infinite series is a widely used tool for numerically solving problems involving FDEs. The weights of the approximation formula are obtained from the coefficients of a generator $W_1(z) = (1 - z)^\alpha$ expressed in a power series form [23]. A shifted form of the difference approximation was proposed in [19] to remedy some stability issues in solving space-fractional diffusion equations. Both difference approximations and their shifted form are known to be of first-order accuracy (see section 2 for details).

For higher-order approximations, Lubich [17] obtained some generators in the form of power of polynomials or rational polynomials. The coefficients of the expansion of the generators provide the coefficients for the approximation with higher-order accuracy without shift. Shifted forms with these coefficients give only first-order approximations regardless of their higher orders in non-shifted forms.

Received: 01 September 2023 ; Accepted: 06 January 2024.

* Corresponding author. Email: nasirh@squ.edu.om .

Recently, generators for higher-order approximations with shifts have been obtained by Nasir and Nafa [22] and applied to fractional diffusion equations with second-order accuracy and a third-order approximation was obtained from the second-order approximation and used for numerical approximations in [21].

The generators of these approximations are usually obtained manually by hand calculations, solving system equations, or by symbolic computations and these processes are specific to the problem at hand.

However, most of the scientific research searching for mathematical patterns is aimed at automating the relevant scientific processes so that the task may be performed for more general problems with minimal human intervention [14]. In this regard, in numerical approximations and solution processes, especially in discretized numerical calculus, it is desirable to have general formulas for the approximation of derivatives that can be automated to be selected in approximation processes.

The present authors [9] have obtained an explicit form for generators that gives approximations for fractional derivatives with shifts retaining their higher orders. This form generalizes the Lubich form with shift and hence the Lubich form becomes a special case with no shift.

Interestingly, this explicit form also gives coefficients for finite difference formulas (FDFs) for classical derivatives as demonstrated in [9]. However, it gives higher-order compact FDFs for the first derivative only. Here, a compact FDF means the FDF which uses a minimum number of function values at the discrete grid points. For higher-order classical derivatives, the resulting FDFs are not compact although they are valid for FDFs.

Usually, the derivation of the weights for the FDFs for classical derivatives involves linear combinations of the Taylor series expansions of a function at various grid points about the point of the derivative. For higher order accuracy requirements, this leads to uncontrollable hand calculations, solving large systems of linear equations, or heavy symbolic computations [13]. Besides, this technique is not suitable for fractional derivatives as the latter involves function values at infinite grid points, due to its non-local nature.

Explicit forms for finite difference formulas for classical derivatives have appeared in the past (see for eg. [1, 4, 6, 7, 11, 12, 16, 24, 26, 27, 30]). However, all of them are focused on the classical derivatives only.

In this paper, we extend the explicit form developed in [9] to a more general unified explicit form that gives more new approximations for fractional derivatives and various finite difference formulas for any classical derivative. The formulation also provides error coefficients of the approximation for the first few terms.

We also present an algorithm to efficiently compute the coefficients of the unified difference formulas. This enables one to compute the coefficients in exact form where necessary and real coefficients with efficiency.

This paper is arranged as follows: We start with preliminary preparations in section 2. The unified explicit form is given in section 3. Efficient computational strategies are described in section 4. Various difference formulas for classical and fractional derivatives are given in section 5 demonstrating the unified nature of the explicit form. Applications of the difference formulas are presented in section 6 and finally, section 7 draws some conclusions.

2. PRELIMINARIES AND TERMINOLOGIES

We list here relevant materials and define terminologies related to the subject of this paper. Let $f(x)$ be a sufficiently smooth function defined on a real domain \mathbb{R} .

Definition 2.1. The left(-) and right(+) Riemann-Liouville (R-L) fractional derivatives of real order $\alpha > 0$ are defined as

$${}^{RL}D_{x-}^{\alpha}f(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_{-\infty}^x \frac{f(\eta)}{(x-\eta)^{\alpha+1-n}} d\eta, \quad (2.1)$$

and

$${}^{RL}D_{x+}^{\alpha}f(x) = \frac{(-1)^n}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_x^{\infty} \frac{f(\eta)}{(\eta-x)^{\alpha+1-n}} d\eta, \quad (2.2)$$

respectively, where $n = [\alpha] + 1$, an integer with $n - 1 < \alpha < n$ and $\Gamma(\cdot)$ denotes the gamma function.



Definition 2.2. The left Caputo fractional derivative of order $\alpha > 0$ is defined analogously to (2.1) by

$${}^C D_{x-}^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_{-\infty}^x \frac{f^{(n)}(\eta)}{(x-\eta)^{\alpha+1-n}} d\eta. \tag{2.3}$$

The right Caputo derivative is defined analogously to (2.2).

Definition 2.3. The left/right Grünwald-Letnikov (G-L) fractional derivatives are given respectively by

$${}^{GL} D_{x\mp}^\alpha f(x) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} f(x \mp kh), \tag{2.4}$$

where the *Grünwald weights*, $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k} = \frac{(-1)^k \Gamma(\alpha+1)}{k! \Gamma(\alpha-k+1)}$ are the coefficients of the series expansion of the *Grünwald generator*

$$W_1(z) = (1-z)^\alpha = \sum_{k=0}^{\infty} g_k^{(\alpha)} z^k.$$

These definitions of fractional derivatives are equivalent under certain smoothness conditions on $f(x)$ [23]. Therefore, we denote them commonly as $D_{x\mp}^\alpha f(x)$ when the conditions are met.

Definition 2.4 (Difference approximations). For the left and right fractional derivatives of order $\alpha > 0$, we define

(1) the *Grünwald approximation* (GA) as

$$\delta_{\pm h}^\alpha f(x) = \frac{1}{h^\alpha} \sum_{k=0}^{\infty} g_k^{(\alpha)} f(x \mp kh), \tag{2.5}$$

(2) the *shifted Grünwald approximation* (ShGA) [19] with shift r as

$$\delta_{\pm h,r}^\alpha f(x) = \frac{1}{h^\alpha} \sum_{k=0}^{\infty} g_k^{(\alpha)} f(x \mp (k-r)h), \tag{2.6}$$

where $h > 0$ is the step size between successive discrete points in the domain referenced from x . Hence, the difference approximations are defined on a *uniform grid* points $x_k = x - kh, k \in \mathbb{Z}$.

The approximation order of both GA and ShGA is one for integer shifts [19].

$$\delta_{\pm h,r}^\alpha f(x) = D_{x\mp}^\alpha f(x) + O(h).$$

For a generalization of the ShGA, we define the following.

Definition 2.5. Let $\{w_k^{(\alpha)}\}$ be a sequence of real numbers with generating function

$$W(z) = \sum_{k=0}^{\infty} w_k^{(\alpha)} z^k.$$

Define a shifted difference formula

$$\Delta_{\pm h,p,r}^\alpha f(x) = \frac{1}{h^\alpha} \sum_{k=0}^{\infty} w_k^{(\alpha)} f(x \mp (k-r)h). \tag{2.7}$$

We say that

(1) $W(z)$ approximates the fractional derivatives $D_{x\mp}^\alpha$ if

$$\lim_{h \rightarrow 0} \Delta_{\pm h,p,r}^\alpha f(x) = D_{x\mp}^\alpha f(x), \tag{2.8}$$

(2) $W(z)$ approximates the fractional derivatives $D_{x\mp}^\alpha$ with order p if

$$\Delta_{\pm h,p,r}^\alpha f(x) = D_{x\mp}^\alpha f(x) + O(h^p). \tag{2.9}$$

We call the difference form (2.7) the *Grünwald type approximations* (GTAs), $W(z)$ the *Grünwald type generator* (GTG) and w_k the *Grünwald type weights* (GTWs).



TABLE 1. Polynomials $P_p(z)$ for Lubich generators $W_p(z) = (P_p(z))^\alpha$.

p	$P_p(z)$	p	$P_p(z)$
1	$1 - z$	4	$\frac{25}{12} - 4z + 3z^2 - \frac{4}{3}z^3 + \frac{1}{4}z^4$
2	$\frac{3}{2} - 2z + \frac{3}{2}z^2$	5	$\frac{137}{60} - 5z + 5z^2 - \frac{10}{3}z^3 + \frac{5}{4}z^4 - \frac{1}{5}z^5$
3	$\frac{11}{6} - 3z + \frac{3}{2}z^2 - \frac{1}{3}z^3$	6	$\frac{147}{60} - 6z + \frac{15}{2}z^2 - \frac{20}{3}z^3 + \frac{15}{4}z^4 - \frac{6}{5}z^5 + \frac{1}{6}z^6$

A set of GTGs for higher-order GTAs was established in Lubich [17] by utilizing the characteristic polynomials of linear multistep methods (LMMs) for classical initial value problems. Particularly, a set of generators in the form of $W_p(z) = (P_p(z))^\alpha$ was obtained from the backward difference LMM for orders $1 \leq p \leq 6$ and is listed in Table 1. For $p = 1$, it gives the Grünwald generator $W_1(z)$.

Definition 2.6. We call the generators given in Table 1 for orders $2 \leq p \leq 6$ as the *Lubich generators* and their weights the *Lubich weights*.

The Lubich generators are to be used without a shift to obtain their respective higher approximation orders. Applying the Lubich weights with non-zero shifts reduces the approximation orders to one [22], [9].

Nasir and Nafa [22] obtained a generator $W_{2,r}(z)$ for an order 2 approximation whose weights can be used with shift without reducing the order:

$$W_{2,r}(z) = \left(\left(\frac{3}{2} - \frac{r}{\alpha} \right) + \left(-2 + \frac{2r}{\alpha} \right) z + \left(\frac{1}{2} - \frac{r}{\alpha} \right) z^2 \right)^\alpha. \tag{2.10}$$

An equivalent characterization for a GTA of order p with shift r was established by the authors in [22] and is given in Proposition 2.7.

Proposition 2.7. (Theorem 1 [21, 22]): Let $\alpha > 0, n = [\alpha] + 1$, and a non-negative integer m be given. Let a function $f(x) \in C^{m+n+1}(\mathbb{R})$ and $D^k f(x) = \frac{d^k}{dx^k} f(x) \in L_1(\mathbb{R})$ for $0 \leq k \leq m + n + 1$. Then, a generator $W(z)$ approximates the fractional derivatives $D_{x\pm}^\alpha f(x)$ with order p and shift $r, 1 \leq p \leq m$, if and only if

$$G(z) = \frac{1}{z^\alpha} W(e^{-z}) e^{rz} = 1 + O(z^p). \tag{2.11}$$

Moreover, if $G(z) = 1 + \sum_{l=p}^\infty a_l z^l$, where $a_l \equiv a_l(\alpha, r)$, then we have

$$\begin{aligned} \Delta_{\pm h,p,r}^\alpha f(x) &= D_{x\pm}^\alpha f(x) + h^p a_p D_{x\pm}^{\alpha+p} f(x) + h^{p+1} a_{p+1} D_{x\pm}^{\alpha+p+1} f(x) + \dots \\ &\quad + h^m a_m D_{x\pm}^{\alpha+m} f(x) + O(h^{m+1}). \end{aligned} \tag{2.12}$$

This characterization theorem was used in [9] to obtain an explicit form for generators of the form $W(z) = (\beta_0 + \beta_1 z + \dots + \beta_p z^p)^\alpha$ for an approximation of order p with shift r .

Proposition 2.8. (Theorem 3 [9]) The generator of the form

$$W_{p,r}(z) = (\beta_0 + \beta_1 z + \dots + \beta_p z^p)^\alpha, \tag{2.13}$$

approximates the fractional derivatives $D_{x\mp}^\alpha f(x)$ with order p and shift r if and only if the coefficients β_j are given by

$$\beta_j = - \left(\sum_{\substack{m=0 \\ m \neq j}}^p \prod_{\substack{l=0 \\ l \neq m, j}}^p (\lambda - j) \right) \left(\prod_{\substack{m=0 \\ m \neq j}}^p \frac{1}{j - m} \right), j = 0, 1, 2, \dots, p, \tag{2.14}$$

where $\lambda = r/\alpha$, and the leading error coefficient is given by



TABLE 2. Coefficients for $p(z) = \beta_0 + \beta_1 z + \dots + \beta_p z^p$ for generator $W_{p,r}(z) = (p(z))^\alpha$.

p	Coefficients $\beta_j, \quad 0 \leq j \leq p$	with $\lambda = r/\alpha$
1	$\beta_0 = 1$	$\beta_1 = -1$
2	$\beta_0 = -\lambda + \frac{3}{2}$ $\beta_2 = -\lambda + \frac{1}{2}$	$\beta_1 = 2\lambda - 2$
3	$\beta_0 = \frac{\lambda^2}{2} - 2\lambda + \frac{11}{6}$ $\beta_2 = \frac{3\lambda^2}{2} - 4\lambda + \frac{3}{2}$	$\beta_1 = -\frac{3\lambda^2}{2} + 5\lambda - 3$ $\beta_3 = -\frac{\lambda^2}{2} + \lambda - \frac{1}{3}$
4	$\beta_0 = -\frac{\lambda^3}{6} + \frac{5\lambda^2}{4} - \frac{35\lambda}{12} + \frac{25}{12}$ $\beta_2 = -\lambda^3 + 6\lambda^2 - \frac{19\lambda}{2} + 3$ $\beta_4 = -\frac{\lambda^3}{6} + \frac{3\lambda^2}{4} - \frac{11\lambda}{12} + \frac{1}{4}$	$\beta_1 = \frac{2\lambda^3}{3} - \frac{9\lambda^2}{2} + \frac{26\lambda}{3} - 4$ $\beta_3 = \frac{2\lambda^3}{3} - \frac{7\lambda^2}{2} + \frac{14\lambda}{3} - \frac{4}{3}$
5	$\beta_0 = \frac{\lambda^4}{24} - \frac{\lambda^3}{2} + \frac{17\lambda^2}{8} - \frac{15\lambda}{4} + \frac{137}{60}$ $\beta_2 = \frac{5\lambda^4}{12} - \frac{13\lambda^3}{3} + \frac{59\lambda^2}{4} - \frac{107\lambda}{6} + 5$ $\beta_4 = \frac{5\lambda^4}{24} - \frac{11\lambda^3}{6} + \frac{41\lambda^2}{8} - \frac{61\lambda}{12} + \frac{5}{4}$	$\beta_1 = -\frac{5\lambda^4}{24} + \frac{7\lambda^3}{3} - \frac{71\lambda^2}{8} + \frac{77\lambda}{6} - 5$ $\beta_3 = -\frac{5\lambda^4}{12} + 4\lambda^3 - \frac{49\lambda^2}{4} + 13\lambda - \frac{10}{3}$ $\beta_5 = -\frac{\lambda^4}{24} + \frac{\lambda^3}{3} - \frac{7\lambda^2}{8} + \frac{5\lambda}{6} - \frac{1}{5}$

$$R = \frac{\alpha}{(p+1)!} \sum_{j=0}^p (\lambda - j)^{p+1} \beta_j. \tag{2.15}$$

Using this explicit form, the authors in [9] have constructed generators of the form (2.13), for orders $1 \leq p \leq 6$ with shifts and the first five of the generators are listed in Table 2.

When there is no shift ($r = 0$), these generators $W_{p,0}(z)$ are the Lubich generators in Table 1. Moreover, for order $p = 1$ it reduces to the Grünwald generator $W_1(z)$.

Definition 2.9. We call the generators given in [9] for orders $2 \leq p \leq 6$ the *Lubich-type generators* and their weights the *Lubich type weights*.

We mention that the Lubich-type generators also give FDFs for classical derivatives with integer derivative order α . However, we point out that the Lubich-type operators give compact finite difference formulas for the first order derivative only. For other higher-order derivatives, we get valid finite difference formulas with non-compact form.

3. A UNIFIED EXPLICIT FORM

In this section, we extend the explicit form (2.14) that appeared in [9] to a more general form that covers compact finite difference formulas for higher order classical derivatives as well as some new Lubich-type generators for fractional derivatives.

For this, we introduce a base differential order d , a positive integer, to express the fractional differential operator as

$$D_{x\pm}^\alpha = (D^d)_{x\pm}^{\frac{\alpha}{d}},$$

and consider approximating the fractional derivative by a Lubich-type generator of the form

$$W(z) = (\beta_0 + \beta_1 z + \dots + \beta_{N-1} z^{N-1})^{\frac{\alpha}{d}} = (P(z))^{\frac{\alpha}{d}}, \tag{3.1}$$

where $P(z)$ corresponds to the classical derivative operator D^d .

The coefficients β_j in (3.1) are to be determined based on the fractional order α , the required approximation order p , and shift r for the GTA. The degree $N - 1$ of $P(z)$ will also be determined according to the choice of p and d . Then, we have the following theorem leading to our main result.

Theorem 3.1. *With assumptions of Proposition 2.7, the generator of the form*



$W(z) = (\beta_0 + \beta_1 z + \dots + \beta_{N-1} z^{N-1})^{\frac{\alpha}{d}}$, where d is a positive integer, approximates the fractional derivatives $D_{x\mp}^{\alpha} f(x)$ at x with order p and shift r if and only if the coefficients β_j satisfy the linear system

$$\sum_{j=0}^{N-1} (\lambda - j)^k \beta_j = d! \delta_{d,k}, \quad k = 0, 1, \dots, N - 1, \tag{3.2}$$

where $\lambda = rd/\alpha$, $N = p + d$ and $\delta_{d,k}$ is the Kronecker delta having a value of one for $k = d$ and zero otherwise.

Proof. In view of Proposition 2.7, we have $G(z) = \frac{1}{z^\alpha} W(e^{-z}) e^{rz} = 1 + O(z^p)$. This gives

$$\begin{aligned} G(z) &= \frac{1}{z^\alpha} \left(\sum_{j=0}^{N-1} \beta_j e^{-jz} \right)^{\frac{\alpha}{d}} e^{rz} = \frac{1}{z^\alpha} \left(\sum_{j=0}^{N-1} \beta_j e^{(rd/\alpha - j)z} \right)^{\frac{\alpha}{d}} = \left(\frac{1}{z^d} \sum_{j=0}^{N-1} \beta_j e^{\lambda_j z} \right)^{\frac{\alpha}{d}} \\ &= \left(\frac{1}{z^d} \sum_{j=0}^{N-1} \beta_j \sum_{k=0}^{\infty} \frac{1}{k!} \lambda_j^k z^k \right)^{\frac{\alpha}{d}} = \left(\frac{1}{z^d} \sum_{k=0}^{\infty} b_k z^k \right)^{\frac{\alpha}{d}} \\ &= \left(\frac{b_0}{z^d} + \frac{b_1}{z^{d-1}} + \dots + \frac{b_{d-1}}{z} + b_d + \sum_{k=d+1}^{\infty} b_k z^{k-d} \right)^{\frac{\alpha}{d}} = 1 + O(z^p), \end{aligned}$$

where $\lambda_j = \lambda - j$, $\lambda = \frac{rd}{\alpha}$ and

$$b_k = \frac{1}{k!} \sum_{j=0}^{N-1} \lambda_j^k \beta_j, \quad k = 0, 1, 2, \dots \tag{3.3}$$

Since $G(z)$ does not have any pole singularities by virtue of (2.11), we have $b_k = 0$ for $k = 0, 1, \dots, d - 1$. Moreover, since $G(0) = 1$, we have $b_d = 1$. These are the consistency conditions for the GTA with generator $W(z)$. Now, for order $p = 1$, these conditions give the system (3.2) with $N = 1 + d$ and the proof ends.

For $p > 1$, $G(z)$ reduces to

$$G(z) = \left(1 + \sum_{k=d+1}^{\infty} b_k z^{k-d} \right)^{\frac{\alpha}{d}} =: (1 + X)^\gamma = 1 + O(z^p),$$

where $\gamma = \frac{\alpha}{d}$ and

$$X = \sum_{k=d+1}^{\infty} b_k z^{k-d}. \tag{3.4}$$

Expansion of $(1 + X)^\gamma$ gives

$$1 + \gamma X + \frac{\gamma(\gamma - 1)}{2!} X^2 + \dots = 1 + O(z^p). \tag{3.5}$$

The term with z appears in the term γX only on the left-hand side of (3.5). This gives $b_{d+1} = 0$. The same is true for $b_k, k = d + 1, d + 2, \dots, p + d - 1$, by successively comparing the coefficients of z^{k-d} to gain $O(z^p)$ in (3.5).

Altogether, we have $b_k = \delta_{d,k}, k = 0, 1, 2, \dots, p + d - 1$ which yields the linear system (3.2) with (3.3) and $N = p + d$. □

The linear system (3.2) can be expressed in matrix form

$$V(\lambda_0, \lambda_1, \dots, \lambda_{N-1}) \mathbf{b} = \mathbf{d}, \tag{3.6}$$



with $\mathbf{b} = [\beta_0, \beta_1, \dots, \beta_{N-1}]^T$, $\mathbf{d} = [0, 0, \dots, d!, 0, \dots, 0]^T$, where $d!$ is at the d^{th} position, and the Vandermonde matrix $V \equiv V(\lambda_0, \lambda_1, \dots, \lambda_{N-1})$ whose columns are $[1, \lambda_k, \lambda_k^2, \dots, \lambda_k^{N-1}]^T, k = 0, 1, \dots, N - 1$, with its determinant

$$|V| = \prod_{0 \leq n < m \leq N-1} (\lambda_m - \lambda_n). \tag{3.7}$$

Our main result is that the coefficients $\beta_j, 0 \leq j < N$ of \mathbf{b} in (3.6) can be explicitly expressed as given in the following theorem.

Theorem 3.2. *Let $\alpha > 0$, a positive integer $d \geq 1$ and $f(x)$ be a sufficiently smooth function such that $D_{x^\pm}^\alpha f(x)$ is defined. For a GTA (2.7) for $D_{x^\pm}^\alpha f(x)$ of order p and shift r with the generator in the form (3.1), the coefficients β_j are given by*

$$\beta_j = \frac{N_j}{D_j}, \quad j = 0, 1, \dots, N - 1, \tag{3.8}$$

where $N = p + d$ and

$$N_j = \sum_{\substack{0 \leq m_1 < m_2 < \dots < m_{p-1} \leq N-1 \\ m_i \neq j, 1 \leq i \leq p-1}} \prod_{k=0}^{p-1} (\lambda - m_k), \quad D_j = \frac{(-1)^d}{d!} \prod_{\substack{m=0 \\ m \neq j}}^{N-1} (j - m). \tag{3.9}$$

Moreover, the leading and some successive error coefficients R_m of the approximation are given by

$$R_m = \frac{\alpha}{m!d} \sum_{j=0}^{N-1} (\lambda - j)^m \beta_j, \quad m = N, N + 1, \dots, N + p. \tag{3.10}$$

Proof. Solving the Vandermonde system (3.6) by Cramer’s rule, we get

$$\beta_j = \frac{|V_j(\mathbf{d})|}{|V|}, \quad j = 0, 1, \dots, N - 1, \tag{3.11}$$

where $|V| = |V(\lambda_0, \lambda_1, \dots, \lambda_{N-1})|$ and $V_j(\mathbf{d})$ is the matrix obtained from V by replacing its j^{th} column with \mathbf{d} .

For evaluating the determinant $|V_j(\mathbf{d})|$, observe that the j^{th} column has only one non-zero entry $d!$ at the d^{th} place. Hence, pivoting with the j^{th} column, we have $|V_j(\mathbf{d})| = (-1)^{j+d} |V_{j,d}|$, where $|V_{j,d}|$ is the determinant of the matrix obtained from removing the j^{th} column and d^{th} row from V , and is given by (see Appendix A)

$$|V_j(\mathbf{d})| = (-1)^{j+d} d! \prod_{\substack{m > n \\ m, n \neq j}} (\lambda_m - \lambda_n) N_j.$$

Rearranging the Vandermonde determinant (3.7) as

$$|V| = \prod_{\substack{0 \leq n < m \leq N-1 \\ m, n \neq j}} (\lambda_m - \lambda_n) (-1)^j \prod_{\substack{m=0 \\ m \neq j}}^{N-1} (\lambda_m - \lambda_j),$$

and substituting in (3.11), we have (3.8) with $\lambda_m - \lambda_j = j - m$.

For the leading and subsequent error terms, note that (3.4) is reduced to

$$X = b_N z^p + b_{N+1} z^{p+1} + \dots + b_{N+p-1} z^{2p-1} + \dots .$$

In (3.5), the terms with $z^k, k = p, p + 1, \dots, 2p - 1$ appear only in X . Thus, the error coefficients are given by

$$R_m = \gamma b_m = \frac{\alpha}{m!d} \sum_{j=0}^{N-1} (\lambda - j)^m \beta_j, \quad N \leq m < N + p.$$

□



The numerator and denominator terms of the coefficients β_j can be easily and efficiently evaluated despite their complicated expressions. Moreover, for rational parameters of α, p and r , these terms will also be rational numbers. Therefore, finite difference coefficients for classical derivatives can be evaluated to their exact rational values.

4. EFFICIENT EVALUATIONS

We describe the details of evaluating the coefficients of the unified formulation. The denominator and numerator terms of the coefficients β_j are evaluated separately.

Separation of the evaluation of numerators and denominators allows one to implement the explicit expression efficiently and to express the coefficients in exact fractional form where possible, that is, when λ is an integer or a rational number.

4.1. Evaluating the denominator. The denominator is evaluated in recursive form for efficient computation as follows: The denominator of the coefficient β_j given in (3.9) can be expressed as

$$D_j = \frac{(-1)^d}{d!} \prod_{m=0}^{N-1} (j-m) = \frac{(-1)^{p-1-j}}{d!} j!(N-1-j)!$$

This can be described by the recursive form

$$D_0 = \prod_{m=d+1}^{N-1} (-m), \quad D_j = \frac{-j}{N-j} D_{j-1}, \quad j = 1, 2, \dots, N-1.$$

Hence, the algorithm for the denominator is as follows:

Algorithm 1. *Compute the denominators*

- (1) *Compute D_0 :*
 $D_0 \leftarrow -m D_0, \quad \text{for } m = d+1, d+2, \dots, N-1.$
- (2) *Compute D_j :*
 $D_j \leftarrow \frac{-j}{N-j} D_{j-1}, \quad \text{for } j = 1, 2, \dots, N-1.$

Since the denominator is independent of any shift r and derivative order α , it can be evaluated once and used for various shifts and order α .

4.2. Evaluating the numerator. An efficient algorithm for evaluating the numerators can be devised once we identify the sum-product term in (3.9) as elementary symmetric polynomials (ESPs) $S(X_j, p-1)$ in the set

$$X_j = \{\lambda_k = \lambda - k : 0 \leq k \leq N-1, k \neq j\}.$$

The ESPs can be efficiently computed by the recursive relations described in Appendix B. We use the operations (B.1) and (B.2) to obtain the required coefficients $S(X_j, p-1), j = 0, 1, \dots, N-1$.

Note that each numerator in β_j has the ESP on distinct sets A_j . In fact, $S(X_j, p-1)$ is the coefficient of $x^{N-1-(p-1)} = x^d$ in the monomial

$$M_j(x) = \prod_{m=0, m \neq j}^{N-1} (x + x_m),$$

of degree $N-1$ that can be expressed in the recursive form

$$M_0(x) = \prod_{m=1}^{N-1} (x + x_m), \quad M_j(x) = M_{j-1}(x) \frac{x + x_{j-1}}{x + x_j}.$$

The polynomials $M_j(x)$ can be computed by combining the two recurrence relations in (B.1) and (B.2) as follows: Let

$$M_{j-1}(x) = \sum_{k=0}^{N-1} p_k x^k, \quad M_j(x) = \sum_{k=0}^{N-1} q_k x^k.$$



and denote their coefficient vectors as \mathbf{p} and \mathbf{q} respectively. Then, the algorithm to compute the numerators is devised as follows:

Algorithm 2. *Computing the numerator terms N_j .*

- (1) Compute $M_0(x)$:
 - $p_0 = 1,$
 - For $m = 1, 2, \dots, N - 1,$ with $p_{-1} = 0,$
 - $p_k \leftarrow p_{k-1} + x_m p_k, \quad k = m, m - 1, \dots, 0.$
 - $N_0 = S(X_0, p - 1) = p_d.$
- (2) Compute $M_j(x)$:
 - For $j = 1, 2, \dots, N - 1,$ with $p_N = q_N = 0,$
 - $q_k \leftarrow p_k + x_{j-1} p_{k+1} - x_j q_{k+1},$ for $k = N - 1, N - 2, \dots, 0.$
 - $\mathbf{p} \leftarrow \mathbf{q}.$
 - $N_j = S(X_j, p - 1) = q_d.$

4.3. Evaluation of Error. The leading and subsequent error coefficients are evaluated directly as the expression involves only one summation. From (3.10), we have the following algorithm:

Algorithm 3. *Computing the error coefficient of order $m, p \leq m \leq 2p.$*

- Given $\beta_j = N_j/D_j$ for $j = 0, 1, \dots, N - 1,$
- (1) $E \leftarrow 0,$
- For $j = 0, 1, \dots, N - 1, E \leftarrow E + \lambda_j^{m+d} \beta_j,$
- (2) $R_m \leftarrow \frac{\alpha}{(m+d)!d} E.$

4.4. Operation count. The direct evaluation of the numerator part in (3.9) computes the sum-product form with all possible combinations. The operation counts for computing the numerator part with base differential order d and approximation order p is MN additions and $MN(p - 1)$ multiplications, where $M = \binom{N-1}{p-1}$ and $N = p + d.$ For small values of $N,$ this count is reasonably small. However, for large $N,$ the operation count uncontrollably increases. For example, the number of additions and multiplications for computing the numerator for a 10-th order derivative with approximation order 10 are 1847560 and 16628040 respectively.

However, the efficient computation described above computes the same numerator in only 968 additions and the same amount of multiplications.

5. DIFFERENCE FORMULAS

We use the unified explicit form in this paper to construct approximating Lubich-type generators with a given base differential order for the left and right fractional derivatives along with their leading error coefficients. We also list some FDFs of various types that can be constructed from the unified formulation.

5.1. Difference approximations for fractional derivatives. When the base differential operator is the first derivative operator $D = d/dx$ with base differential order $d = 1,$ we obtain the Lubich-type generators obtained in [22] with shift r and are given in Table 2. When there is no shift ($r = 0$), this gives the Lubich generators in Table 1. Hence, our unified formulation is a further generalization of the Lubich-type generators and the Lubich generators for the approximation of fractional derivatives with and without shifts respectively.

When the base differential operator is of order $d > 1,$ we get new approximating generators. Tables 3 and 4 list some approximating generators with base differential orders $d = 2$ and 3 respectively and approximation orders $p = 1, 2, 3, 4, 5.$ Note that the first order approximations in all cases turn out to be the same Grünwald generator $W_1(z) = (1 - z)^\alpha.$



TABLE 3. Coefficients for polynomial $p(z) = \beta_0 + \beta_1 z + \dots + \beta_{p+d-1} z^{p+d-1}$ for generator $W_{p,d}(z) = (p(z))^{\alpha/d}$ with base differential order $d = 2$ and $\lambda = rd/\alpha$.

p	Coefficients $\beta_j, 0 \leq j \leq p + d - 1$	with $d = 2$ and $\lambda = rd/\alpha$
1	$\beta_0 = 1$ $\beta_2 = 1$	$\beta_1 = -2$
2	$\beta_0 = -\lambda + 2$ $\beta_2 = -3\lambda + 4$	$\beta_1 = 3\lambda - 5$ $\beta_3 = \lambda - 1$
3	$\beta_0 = \frac{\lambda^2}{2} - \frac{5\lambda}{2} + \frac{35}{12}$ $\beta_2 = 3\lambda^2 - 12\lambda + \frac{19}{2}$ $\beta_4 = \frac{\lambda^2}{2} - \frac{3\lambda}{2} + \frac{11}{12}$	$\beta_1 = -2\lambda^2 + 9\lambda - \frac{26}{3}$ $\beta_3 = -2\lambda^2 + 7\lambda - \frac{14}{3}$
4	$\beta_0 = -\frac{\lambda^3}{6} + \frac{3\lambda^2}{2} - \frac{17\lambda}{4} + \frac{15}{4}$ $\beta_2 = -\frac{5\lambda^3}{3} + 13\lambda^2 - \frac{59\lambda}{2} + \frac{107}{6}$ $\beta_4 = -\frac{5\lambda^3}{6} + \frac{11\lambda^2}{2} - \frac{41\lambda}{4} + \frac{61}{12}$	$\beta_1 = \frac{5\lambda^3}{6} - 7\lambda^2 + \frac{71\lambda}{4} - \frac{77}{6}$ $\beta_3 = \frac{5\lambda^3}{3} - 12\lambda^2 + \frac{49\lambda}{2} - 13$ $\beta_5 = \frac{\lambda^3}{6} - \lambda^2 + \frac{7\lambda}{4} - \frac{5}{6}$
5	$\beta_0 = \frac{\lambda^4}{24} - \frac{7\lambda^3}{12} + \frac{35\lambda^2}{12} - \frac{49\lambda}{8} + \frac{203}{45}$ $\beta_2 = \frac{5\lambda^4}{8} - \frac{95\lambda^3}{12} + \frac{137\lambda^2}{4} - \frac{461\lambda}{8} + \frac{117}{4}$ $\beta_4 = \frac{5\lambda^4}{8} - \frac{85\lambda^3}{12} + \frac{107\lambda^2}{4} - \frac{307\lambda}{8} + \frac{33}{2}$ $\beta_6 = \frac{\lambda^4}{24} - \frac{5\lambda^3}{12} + \frac{17\lambda^2}{12} - \frac{15\lambda}{8} + \frac{137}{180}$	$\beta_1 = -\frac{\lambda^4}{4} + \frac{10\lambda^3}{3} - \frac{31\lambda^2}{2} + 29\lambda - \frac{87}{5}$ $\beta_3 = -\frac{5\lambda^4}{6} + 10\lambda^3 - \frac{121\lambda^2}{3} + 62\lambda - \frac{254}{9}$ $\beta_5 = -\frac{\lambda^4}{4} + \frac{8\lambda^3}{3} - \frac{19\lambda^2}{2} + 13\lambda - \frac{27}{5}$

The coefficients w_k of the generators can be computed from the J.C.P. Miller formula given as follows (see [29] for a proof):

$$W(z) = \left(\sum_{j=0}^{N-1} \beta_j z^j \right)^\gamma = \sum_{k=0}^{\infty} w_k z^k,$$

$$w_0 = \beta_0^\gamma, \quad w_m = \frac{1}{mw_0} \sum_{k=1}^M (k(\gamma + 1) - m) \beta_k w_{m-k}, \quad m = 1, 2, \dots, \tag{5.1}$$

where $M = \min(m, N - 1)$.

The Grünwald type approximation for the left or right fractional derivative is then given by (2.7).

5.2. Finite difference formulas. The FDFs for classical derivatives come in various shapes and flavors. Forward, backward, and central finite differences are well-known. There are also finite difference forms that give derivatives at any specified grid point. We call them *shifted finite difference forms*. In addition, there are finite difference forms that give derivatives at a point between two adjacent points. These are called staggered finite difference forms. There are compact finite difference forms that use the minimum number of grid points for a specified order of accuracy. In contrast, there are also non-compact finite difference forms.

Our proposed algorithm gives all these forms for specified choices of input parameters of d, p , and r . In the following, we list the input parameters and their output finite difference forms in some detail.

5.3. Compact finite difference forms. Compact finite difference forms are those that use the minimum number of grid points for a specified derivative order and order of approximation and are commonly used in applications. There are known as compact finite difference forms such as forward (right), backward (left), and central (symmetric). There are also other compact forms in shifted and staggered formulations. The presented algorithm gives all these forms.

In the presented algorithm, the input parameters for compact finite difference forms for a classical derivative α are chosen to be equal to the base order d . The required order of approximation can be independently chosen. The



TABLE 4. Coefficients for polynomial $p(z) = \beta_0 + \beta_1 z + \dots + \beta_{p+d-1} z^{p+d-1}$ for generator $W_{p,d}(z) = (p(z))^{\alpha/d}$ with base differential order $d = 3$ and $\lambda = rd/\alpha$.

p	Coefficients $\beta_j, 0 \leq j \leq p + d - 1$	with $d = 3$ and $\lambda = rd/\alpha$
1	$\beta_0 = 1$ $\beta_2 = 3$	$\beta_1 = -3$ $\beta_3 = -1$
2	$\beta_0 = -\lambda + \frac{5}{2}$ $\beta_2 = -6\lambda + 12$ $\beta_4 = -\lambda + \frac{3}{2}$	$\beta_1 = 4\lambda - 9$ $\beta_3 = 4\lambda - 7$
3	$\beta_0 = \frac{\lambda^2}{2} - 3\lambda + \frac{17}{4}$ $\beta_2 = 5\lambda^2 - 26\lambda + \frac{59}{2}$ $\beta_4 = \frac{5\lambda^2}{2} - 11\lambda + \frac{41}{4}$	$\beta_1 = -\frac{5\lambda^2}{2} + 14\lambda - \frac{71}{4}$ $\beta_3 = -5\lambda^2 + 24\lambda - \frac{49}{2}$ $\beta_5 = -\frac{\lambda^2}{2} + 2\lambda - \frac{7}{4}$
4	$\beta_0 = -\frac{\lambda^3}{6} + \frac{7\lambda^2}{4} - \frac{35\lambda}{6} + \frac{49}{8}$ $\beta_2 = -\frac{5\lambda^3}{2} + \frac{95\lambda^2}{4} - \frac{137\lambda}{2} + \frac{461}{8}$ $\beta_4 = -\frac{5\lambda^3}{2} + \frac{85\lambda^2}{4} - \frac{107\lambda}{2} + \frac{307}{8}$ $\beta_6 = -\frac{\lambda^3}{6} + \frac{5\lambda^2}{4} - \frac{17\lambda}{6} + \frac{15}{8}$	$\beta_1 = \lambda^3 - 10\lambda^2 + 31\lambda - 29$ $\beta_3 = \frac{10\lambda^3}{3} - 30\lambda^2 + \frac{242\lambda}{3} - 62$ $\beta_5 = \lambda^3 - 8\lambda^2 + 19\lambda - 13$
5	$\beta_0 = \frac{\lambda^4}{24} - \frac{2\lambda^3}{3} + \frac{23\lambda^2}{6} - \frac{28\lambda}{3} + \frac{967}{120}$ $\beta_2 = \frac{7\lambda^4}{8} - 13\lambda^3 + \frac{135\lambda^2}{2} - 142\lambda + \frac{3929}{40}$ $\beta_4 = \frac{35\lambda^4}{24} - 20\lambda^3 + \frac{565\lambda^2}{6} - 176\lambda + \frac{2545}{24}$ $\beta_6 = \frac{7\lambda^4}{24} - \frac{11\lambda^3}{3} + \frac{95\lambda^2}{6} - \frac{82\lambda}{3} + \frac{1849}{120}$	$\beta_1 = -\frac{7\lambda^4}{24} + \frac{9\lambda^3}{2} - \frac{295\lambda^2}{12} + \frac{111\lambda}{2} - \frac{638}{15}$ $\beta_3 = -\frac{35\lambda^4}{24} + \frac{125\lambda^3}{6} - \frac{1235\lambda^2}{12} + \frac{1219\lambda}{6} - \frac{389}{3}$ $\beta_5 = -\frac{7\lambda^4}{8} + \frac{23\lambda^3}{2} - \frac{207\lambda^2}{4} + \frac{185\lambda}{2} - \frac{268}{5}$ $\beta_7 = -\frac{\lambda^4}{24} + \frac{\lambda^3}{2} - \frac{25\lambda^2}{12} + \frac{7\lambda}{2} - \frac{29}{15}$

TABLE 5. Choice of shift parameters for compact finite difference forms.

$\alpha = d$	Left	Right	Central	Shifted	Staggered
Shift r	0	$p + d - 1$	$(p + d - 1)/2$	any int	any real

number of grid points for all the compact finite difference forms for a differential order d and approximation order p is $N = p + d$. All the different forms are obtained through the choice of the shift parameter r as given in Table 5.

Table 6 lists some compact finite difference forms with one example of each kind mentioned above. The coefficient of the function at the point where the derivative is intended is indicated within parentheses. This is given by the index $c = r - [r] =$ the fractional part of r . The finite difference summation formula starts with function index r , (that is f_r) from the leftmost coefficient and decrements by one for each subsequent coefficient. The sum is then divided by h^α .

The error term takes the leading error coefficient from the explicit form for the error multiplied by $h^p f^{(p)}(\xi)$, where ξ is a point between the grid points.

For example, the central compact finite difference form for the derivative of order $\alpha = 3$ with base $d = 3$ and approximation order $p = 4$ and shift $r = 3$ in Table 6 is read as

$$\frac{d^3}{dx^3} f_0 = \frac{1}{h^3} \left(-\frac{1}{8} f_3 + f_2 - \frac{13}{8} f_1 + \frac{13}{8} f_{-1} - f_{-2} + \frac{1}{8} f_{-3} \right) - \frac{7}{120} h^4 f^{(4)}(\xi),$$

where $f_k = f(x_c + kh)$ and c is the fractional part of r .

Note that $0 \leq c < 1$ with non-zero value for non-integer shift r – the staggered case and zero for integer shifts. Thus, the last row of Table 6 is read as

$$\frac{d^2}{dx^2} f_{0.5} = \frac{1}{h^2} \left(\frac{3}{16} f_{1.5} + \frac{41}{48} f_{0.5} - \frac{67}{24} f_{-1.5} + \frac{19}{8} f_{-2.5} - \frac{35}{48} f_{-3.5} + \frac{5}{48} f_{-4.5} \right) + \frac{341}{5760} h^4 f^{(4)}(\xi).$$



TABLE 6. Some Compact finite difference forms.

d	α	p	r	Coefficients for Compact forms	Error	Names
1	1	3	0	$(\frac{11}{6}), -3, \frac{3}{2}, -\frac{1}{3}$	$-\frac{1}{4}$	Left
3	3	4	3	$-\frac{1}{8}, 1, -\frac{13}{8}, (0), \frac{13}{8}, -1, \frac{1}{8}$	$-\frac{7}{120}$	Central
2	2	4	1	$\frac{5}{6}, (-\frac{5}{4}), -\frac{1}{3}, \frac{7}{6}, -\frac{1}{2}, \frac{1}{12}$	$\frac{13}{180}$	Shifted
3	3	4	6	$-\frac{15}{8}, 13, -\frac{307}{8}, 62, -\frac{461}{8}, 29, (-\frac{49}{8})$	$-\frac{29}{15}$	Right
2	2	4	1.5	$\frac{3}{16}, (\frac{41}{48}), -\frac{67}{24}, \frac{19}{8}, -\frac{35}{48}, \frac{5}{48}$	$\frac{341}{5760}$	Staggered

5.4. Non-compact finite difference forms. In contrast to the compact finite difference forms, non-compact finite difference forms use more grid points than that of compact forms. All the various finite difference forms described in compact forms are also available in the non-compact forms with the same choices of the shift r .

The non-compact finite difference forms are obtained by choosing the base order d such that $\alpha = \gamma d$ for some integer $\gamma > 1$. The coefficients β_j for the polynomial $p(z) = \sum_{j=0}^{p+d-1} \beta_j z^j$ are first computed by using the unified explicit form in (3.9). The generator for the non-compact form is then given by $W(z) = (p(z))^\gamma$ and is expanded to obtain the non-compact coefficients. The expansion can be performed by the J. C. P. Miller recurrence formula as was done in the fractional derivative case.

For example, the non-compact finite difference formula for the second derivative ($\alpha = 2$) of approximation order $p = 3$ with $d = 1 \neq \alpha$ and shift $r = 1$ has the coefficients and leading error coefficient computed from the unified formulation as

$$\{\beta_k\} = \left(\frac{23}{24}, -\frac{7}{8}, -\frac{1}{8}, \frac{1}{24} \right), \quad R_3 = \frac{1}{12},$$

respectively. Therefore, the generator for the approximate formula is given with power $\gamma = \alpha/d = 2$ as

$$\begin{aligned} W(z) &= \left(\frac{23}{24} - \frac{7}{8}z - \frac{1}{8}z^2 + \frac{1}{24}z^3 \right)^2 \\ &= \frac{529}{576} - \frac{161}{96}z + \frac{101}{192}z^2 + \frac{43}{144}z^3 - \frac{11}{192}z^4 - \frac{1}{96}z^5 + \frac{1}{576}z^6. \end{aligned}$$

Hence, the non-compact finite difference formula with its error term is given by

$$\begin{aligned} \frac{d^2}{dx^2} f_0 &= \frac{1}{h^2} \left(\frac{529}{576} f_1 - \frac{161}{96} f_0 + \frac{101}{192} f_{-1} \right. \\ &\quad \left. + \frac{43}{144} f_{-2} - \frac{11}{192} f_{-3} - \frac{1}{96} f_{-4} + \frac{1}{576} f_{-5} \right) + \frac{1}{12} h^3 f^{(3)}(\xi). \end{aligned}$$

6. NUMERICAL TESTS

In order to demonstrate the usefulness of the presented unified explicit formulation, we present here some numerical examples. Analyses of these applications are beyond the scope of this paper and will be considered in the future.

6.1. Boundary value problem. Consider the boundary value problem:

$$\frac{d^2}{dx^2} u(x) = f(x), \quad a \leq x \leq b, \quad u(a) = u_a, \quad u(b) = u_b. \quad (6.1)$$



Consider the computational domain $[a, b]_h = \{x_i = a + ih\}_{i=0}^{N-1}$ with uniform discretization of the domain $[a, b]$ with subinterval size $h = (b - a)/(N - 1)$. Usually, the second derivative is approximated by the second-order central difference formula for each internal point. That is, with $u_i = u(x_i)$ and $f_i = f(x_i)$,

$$\frac{d^2}{dx^2}u(x_i) = f(x_i), \quad i = 1, 2, \dots, N - 2, \tag{6.2}$$

is approximated by

$$\frac{1}{h^2} (u_{i-1} - 2u_i + u_{i+1}) = f_i + O(h^2), \quad i = 1, 2, \dots, N - 2. \tag{6.3}$$

The resulting tridiagonal matrix equation is solved, after imposing the boundary conditions, for the discrete solution u_i which approximates $u(x_i)$ with $O(h^2)$ error.

Now that we have the explicit form (3.8) for any order of approximation p with $N = p + d$, we may approximate (6.2) by a higher order approximation through (2.7). Since $\alpha = 2$, we choose $d = 2$, so that $\alpha/d = 1$ and the resulting finite difference formulas with shifts are compact and can be computed for each shift r from (3.1),

$$W(z) = \beta_0 + \beta_1 z + \dots + \beta_{N-1} z^{N-1}, \tag{6.4}$$

where $\beta_j = \beta_j(r), j = 0, 1, \dots, N - 1$. Each equation in (6.2) is approximated by (6.4) with shift $r = i$.

$$\frac{1}{h^2} \sum_{j=0}^{N-1} \beta_j(i) u_j = f_i + O(h^{N-d}), \quad i = 1, 2, \dots, N - 2. \tag{6.5}$$

Note that the order of this approximation is $p = N - d$ and hence, the higher the number of nodes, the higher the order of approximation. The matrix form of this system is given by $BU = F$, where we denote the approximation of u_k as $\hat{u}_k, U = [\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1}]^T, F = [f_1, f_2, \dots, f_{N-2}]^T$, and the matrix B is given by

$$B_{i,j} = \beta_j(i), \quad i = 1, 2, \dots, N - 2, \quad j = 0, 1, \dots, N - 1.$$

The entries of the matrix B are given by (3.8) and are automatically computed by the algorithms for β_j . Imposing boundary conditions, the system reduces to

$$\hat{B}\hat{U} = F - B_0 u_0 - B_{N-1} u_{N-1}, \tag{6.6}$$

where $\hat{B} = [B_1, B_2, \dots, B_{N-2}], B_k, 1 \leq k \leq N - 2$, are the internal columns of B and $\hat{U} = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{N-2}]^T$. Solving (6.6) yields the approximate solution.

Example 6.1. We apply the above numerical schemes to the following boundary value problem:

$$\begin{aligned} \frac{d^2}{dx^2}u(x) &= -\sin(x), \quad -1 < x < 1, \\ u(-1) &= \sin(-1), \quad u(1) = \sin(1). \end{aligned} \tag{6.7}$$

for which the exact solution is $u(x) = \sin(x)$.

In order to compare and see the effectiveness of the automation and accuracy of the unified explicit form, the problem is solved first by using the well-known central difference formula of order 2 for the second derivative and then by using the highest-order compact finite difference formulas that use the full set of nodal values of the solution vector.

Table 7 shows the maximum absolute errors for the central difference method (6.3) and the proposed method in (6.5). As the order of the proposed method increases with the size of the discrete domain, precision errors occur for higher order approximations as the accuracy of the solution surpasses the machine precision—see orders 30, 62, and 126 in Table 7. In order to make sure that these errors are indeed due to accuracy overwhelming the machine precision, we computed the same methods with high precision arithmetic with 300 decimal digits accuracy. In this case, the errors disappear and the absolute maximum errors continue to reduce.



TABLE 7. Example 6.1: Order 2 Central difference and the proposed method.

$N - 1$	Step size h	Order 2 Central	Proposed Higher Order formulas		
			Order	Prec. 15 digits	Prec. 300 digits
4	0.5	2.3370e-01	3	0.00123815	0.00123815
8	0.25	5.3029e-02	7	1.85125e-07	1.85125e-07
16	0.125	1.2951e-02	15	5.55112e-15	2.02095e-17
32	0.0625	3.2190e-03	31	4.71049	7.17654e-42
64	0.03125	8.0358e-04	63	(Precision error)	4.76176e-100
128	0.015625	2.0082e-04	127	(Precision error)	2.71135e-235

TABLE 8. Maximum errors and convergence orders for Example 6.2 using the unified explicit formula with $d = 2, p = 2$.

N	$\alpha = 1.33$		$\alpha = 1.34$		$\alpha = 1.6$	
	Error	Order	Error	Order	Error	Order
8	2.7809e-02	–	2.7809e-02	–	1.7798e-02	–
16	6.1767e-03	2.1707e+00	5.7018e-03	2.1871e+00	4.4935e-03	7.0856e-05
32	1.4568e-03	2.0840e+00	1.3175e-03	2.1136e+00	1.1292e-03	1.9925e+00
64	3.5663e-04	2.0303e+00	3.1401e-04	2.0690e+00	2.8309e-04	1.9960e+00
128	9.3875e-05	1.9256e+00	7.6700e-05	2.0335e+00	7.0856e-05	1.9983e+00
256	5.7965e-05	6.9557e-01	1.9031e-05	2.0109e+00	1.7725e-05	1.9991e+00
512	1.6220e-05	1.8374e+00	4.7521e-06	2.0017e+00	4.4327e-06	1.9996e+00
1024	3.7781e-06	2.1021e+00	1.1880e-06	2.0001e+00	1.1083e-06	1.9998e+00
2048	1.3291e-06	1.5072e+00	2.9696e-07	2.0002e+00	2.7710e-07	1.9999e+00
4096	1.8479e+05	-3.7017e+01	7.4056e-08	2.0036e+00	6.9267e-08	2.0002e+00

Although we solve the resulting matrix equation by the standard classical solvers, the solution process may be efficiently performed seeing that matrix B has a centro-symmetric structure. This has not been considered here and is thus open.

6.2. Fractional boundary value problem. The use of the difference approximations for fractional derivatives is an ongoing research. Therefore, the use of the unified explicit formulation is yet to be explored. Nevertheless, we present here a simple example of a fractional boundary value problem that uses the difference formula with base differential order $d = 2$.

Example 6.2. Consider the fractional boundary value problem

$${}^C D_{x-}^\alpha y(x) = \frac{\Gamma(4+x)}{6} x^3, \quad 0 < x < 1, \quad 1 < \alpha < 2, \quad (6.8)$$

$$u(0) = 0, \quad u(1) = 1.$$

The exact solution is $y(x) = x^{3+\alpha}$.

The domain $[0, 1]$ is discretized with points $x_j = jh, j = 0, 1, \dots, N$ and the fractional derivative is approximated by the order $p = 2$ approximation with base order $d = 2$ and shift $r = 1$ from Table 3. The weights w_k are computed by using the J.C.P. Miller formula in (5.1). The resulting system

$$BY = F,$$

is solved after imposing the boundary conditions.

Table 8 lists the maximum errors and the computed convergence orders for $\alpha = 1.33, 1.34, 1.6$. Note that the computed convergence order for $\alpha = 1.33$ is not up to the theoretical order 2.



The order 2 accuracy is obtained for $\alpha > 4/3$ and has been confirmed by numerical tests with more α values. This can be justified by the convergence of the series of the generator as follows. The generator $W_{2,2}(z)$ for $p = 2, d = 2$ is given by (see Table 3)

$$W_{2,2}(z) = (\beta_0 + \beta_1 z + \beta_2 z^2 + \beta_3 z^3)^{\alpha/2} = (1 - z)^\alpha (\beta_0 + \beta_3 z)^{\alpha/2},$$

where $\beta_0 = 2 - \lambda, \beta_1 = -5 + 3\lambda, \beta_2 = 4 - 3\lambda, \beta_3 = -1 + \lambda$ with $\lambda = rd/\alpha = 2r/\alpha$. The series expansion of $W_{2,2}(z)$ converges when the series of both factors $(1 - z)^\alpha$ and $(\beta_0 + \beta_3 z)^{\alpha/2}$ converge – that is, when $|z| \leq 1$ and $|\beta_3/\beta_0 z| < 1$.

These give $|\beta_3/\beta_0| < 1$ and resolve to $4r/3 < \alpha$. Moreover, for real-valued expansion of the second factor, it requires $\beta_0 > 0$ which leads to $r < \alpha$. For $r = 1$, it follows that $\alpha > 4/3$.

6.3. Fractional initial value problem. We present an example to demonstrate an advantage of the new Grünwald type formulation in the stability of the fractional backward difference formula (FDBF) methods for solving the fractional initial value problems (FIVP).

Consider the FIVP

$${}^C D_t^\alpha y(t) = f(y, y(t)), \quad t \geq 0, \quad n - 1 < \alpha < n, \tag{6.9}$$

$$y(0) = y'(0) = \dots = y^{(n-1)}(0) = 0, \tag{6.10}$$

where $n = [\alpha] + 1$, often $n = 1$ or 2 , the fractional derivative is in the Caputo sense which is equivalent to the R-L fractional derivative owing to the homogeneous initial conditions.

For a uniform discretization $t_j = \tau j, j = 0, 1, \dots, N$ of the time domain $[0, T]$ with $\tau = T/N$, the fractional derivative is approximated by a left Grünwald type approximation through a generator $W(z)$ at $t = t_n$ resulting in the implicit FBDF

$$\frac{1}{\tau^\alpha} \sum_{k=0}^{\infty} w_k^{(\alpha)} y_{n-k} = f_n, \tag{6.11}$$

where y_n is an approximation of $y(t_n), f_n = f(t_n, y_n)$ with the assumption $y_k = 0$ for $k \leq 0$.

For the stability of the FBDFs, we apply the scheme (6.11) to the problem (6.9) with $f(t, y) = \lambda y$, where $\lambda \in \mathbb{C}$.

The stability region for the implicit FBDF is given by

$$S = \{W(\xi) : |\xi| \geq 1\}.$$

The readers are directed to [17], [8] for more information on FBDF and its stability.

Now, we consider two second-order GTAs to compare their stabilities: One is from the Lubich generator $W_2(z)$ given in Table 1 which can also be obtained from the proposed unified formulation with $d = 1$ and $r = 0$. For the other, we consider a new generator $W_{2,2}(z)$ obtained from the unified formulation from Table 3 with $d = 2$ and $r = 0$. They are respectively given by

$$W_2(z) = \left(\frac{3}{2} - 2z + \frac{1}{2}z^2\right)^\alpha \quad \text{and} \quad W_{2,2}(z) = (2 - 5z + 4z^2 - z^3)^{\frac{\alpha}{2}}.$$

The stability regions for these methods are shown in Figures 1 and 2 with the same axes scales for some values of fractional order $0 < \alpha < 1$ and $1 < \alpha < 2$ respectively. The stable regions are outside of the closed regions for each α value.

It can be seen from the figures that the FBDF corresponding to the new generator with $d = 2$ has smaller unstable regions compared to its Lubich counterpart. This shows an improvement in the stability region of the new FBDF method.

7. CONCLUSION

A unified explicit form for the difference formulas for classical and fractional order derivatives was presented. This formulation gives various finite difference formulas such as left, right, central, shifted, compact, and staggered finite difference formulas of an arbitrary order for classical derivatives. The formulation also gives Grünwald type difference approximation of any order for fractional derivatives.



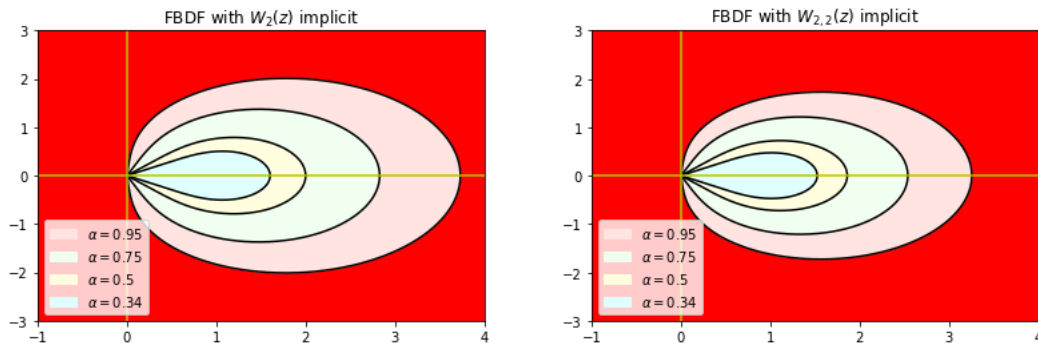


FIGURE 1. Stability regions for $W_2(z), W_{2,2}(z)$ with $0 < \alpha < 1$.

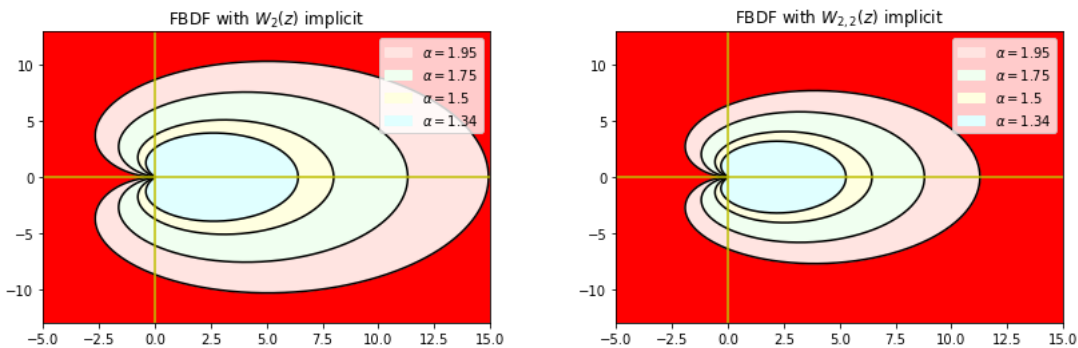


FIGURE 2. Stability regions for $W_2(z), W_{2,2}(z)$ with $1 < \alpha < 2$.

From this unified formulation, some new types of difference formulas for fractional derivatives were also obtained. Basic examples of applications were presented to demonstrate the effect of these difference formulas for classical and fractional boundary and initial value problems. More properties, analyses, and applications of these formulas are open for research.

We believe that the unified explicit formula will contribute to the automation of the solution process of differential problems with minimal manual intervention.

APPENDIX A. VANDEMONDE TYPE DETERMINANTS

In this appendix, we establish the determinant of the matrix obtained in section 3.

Lemma A.1. *It is known that, for a finite sequence of $q + 1$ parameters x_0, x_1, \dots, x_q , the determinant of the Vandermonde matrix $V(x_0, x_1, \dots, x_q)$ of size $(q + 1)$ is given by*

$$|V| = |V(x_0, x_1, \dots, x_q)| = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & \dots & x_q \\ x_0^2 & x_1^2 & x_2^2 & \dots & x_q^2 \\ \dots & \dots & \dots & \dots & \dots \\ x_0^q & x_1^q & x_2^q & \dots & x_q^q \end{vmatrix} = \prod_{0 \leq i < j \leq q} (x_j - x_i).$$



Lemma A.2. Let a variant of the Vandermonde matrix with $q + 1$ parameters be defined as

$$U_k(x_0, x_1, \dots, x_q) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_q \\ x_0^2 & x_1^2 & \dots & x_q^2 \\ \dots & \dots & \dots & \dots \\ x_0^{k-1} & x_1^{k-1} & \dots & x_q^{k-1} \\ x_0^{k+1} & x_1^{k+1} & \dots & x_q^{k+1} \\ \dots & \dots & \dots & \dots \\ x_0^{q+1} & x_1^{q+1} & \dots & x_q^{q+1} \end{bmatrix},$$

where, from the k -th row onward, the powers of the parameters are increased by one. Then, the determinant of U_k , which is of size $(q + 1)$, is given by

$$|U_k| = \prod_{0 \leq i < j \leq q} (x_j - x_i) \left(\sum_{\sigma \in B} \prod_{l \in \sigma} x_l \right) = |V| \left(\sum_{\sigma \in B} \prod_{l \in \sigma} x_l \right), \tag{A.1}$$

where B is the set of all combinations of $q + 1 - k$ elements chosen from the index set $A = \{0, 1, 2, \dots, q\}$ given by $B = \{\sigma = (m_1, m_2, \dots, m_{q+1-k}) : m_k \in A\}$.

Proof. It is clear that $|U_k|$ is a homogeneous polynomial of total degree $1 + 2 + \dots + (k - 1) + (k + 1) + \dots + (q + 1) = (q + 1)(q + 2)/2 - k$. Moreover, for $i \neq j$, by replacing x_i with x_j , we have $|U_k| = 0$ and thus $(x_i - x_j)$ are factors of $|U_k|$. There are $q(q + 1)/2$ such factors as in the Vandermonde determinant $|V|$. Hence,

$$|U_k| = \prod_{0 \leq i < j \leq q} (x_j - x_i) P(x_0, x_1, \dots, x_q),$$

where P is a homogeneous polynomial of total degree $(q + 1)(q + 2)/2 - k - q(q + 1)/2 = q + 1 - k$.

Since there are $q + 1$ parameters, P will have $\binom{q + 1}{q + 1 - k}$ terms of products of combinations of $q + 1 - k$ elements chosen from the $q + 1$ parameters. Thus, P has the form

$$P = C \left(\sum_{\sigma \in A} \prod_{l \in \sigma} x_l \right).$$

The homogeneity of P forces the constant coefficients of the product terms to be the same. Equating one of the terms of $|U_k|$, we see that $C = 1$. □

Lemma A.3. The determinant $|V_j(\mathbf{b})|$ in (3.11) of Theorem 3.2 is given by

$$|V_j(\mathbf{d})| = (-1)^{j+d} d! \prod_{\substack{m > n \\ m, n \neq j}} (\lambda_m - \lambda_n) N_j,$$

where N_j is given in (3.9).

Proof. Since the matrix $V_j(\mathbf{d})$ is obtained from the Vandermonde matrix V by replacing the j^{th} column by the vector \mathbf{d} , the parameter λ_j is removed from the list $\{\lambda_0, \lambda_2, \dots, \lambda_{N-1}\}$. Performing the determinant evaluation with the j^{th}



column, we have

$$|V_j(\mathbf{d})| = (-1)^{j+d} d! \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \lambda_0 & \lambda_1 & \cdots & \lambda_q \\ \lambda_0^2 & \lambda_1^2 & \cdots & \lambda_q^2 \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_0^{d-1} & \lambda_1^{d-1} & \cdots & \lambda_q^{d-1} \\ \lambda_0^{d+1} & \lambda_1^{d+1} & \cdots & \lambda_q^{d+1} \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_0^{q+1} & \lambda_1^{q+1} & \cdots & \lambda_q^{q+1} \end{vmatrix}.$$

The determinant is in the form of the variant determinant in Lemma 2. Hence, we have

$$\begin{aligned} |V_j(\mathbf{d})| &= (-1)^{j+d} d! |V| \left(\sum_{\sigma \in B} \prod_{l \in \sigma} \lambda_l \right) \\ &= (-1)^{j+d} d! |V| \sum_{\substack{0 \leq m_1 < m_2 < \cdots < m_{p-1} \leq N-1 \\ m_i \neq j, 1 \leq i \leq p-1}} \prod_{k=0}^{p-1} \lambda_{m_k}, \end{aligned}$$

where σ are the combinations of $N - 1 - d = p - 1$ elements from the $N - 1$ parameter set $A_j = \{\lambda_m : 0 \leq m \leq N - 1\} \setminus \{\lambda_j\}$.

Now, the last sum-product term is N_j and $|V| = \prod_{\substack{m > n \\ m, n \neq j}} (\lambda_m - \lambda_n)$ ends the proof. □

APPENDIX B. ELEMENTARY SYMMETRIC POLYNOMIALS

The elementary symmetric polynomials on a finite set X of parameters are defined as follows:

Definition B.1. Let $X = \{x_0, x_1, \dots, x_{N-1}\}$ be a set of N parameters and B_k be the set of all combinations σ of k parameters chosen from X . Then

$$S(X, k) = \sum_{\sigma \in B_k} \prod_{l \in \sigma} x_l = \sum_{0 \leq i_1 < i_2 < \cdots < i_k \leq N-1} x_{i_1} x_{i_2} \cdots x_{i_k}.$$

It is interesting to note that the monomial

$$L(x) = \prod_{k=0}^{N-1} (x + x_k),$$

has the expanded form given by

$$L(x) = \sum_{k=0}^N S(X, k) x^{N-k},$$

with $S(X, 0) = 1$. The elementary symmetric polynomials can thus be obtained from the coefficients of powers of the expanded $L(x)$.

The product in $L(x)$ can be performed recursively by multiplying the factors $(x + x_k)$ successively as follows:

- (1) Let $L_{-1}(x) = 1$.
- (2) Define $L_k(x) = L_{k-1}(x)(x + x_k)$, $k = 0, 1, \dots, N - 1$.
- (3) If $L_{k-1}(x) = p_0 + p_1x + \cdots + p_{k-1}x^{k-1}$ and $L_k(x) = q_0 + q_1x + \cdots + q_kx^k$, then, by setting $p_{-1} = p_k = 0$, the coefficients q_m are related by

$$q_m = p_{m-1} + x_k p_m, \quad m = 0, 1, 2 \cdots, k. \tag{B.1}$$



Conversely, one can obtain $L_{k-1}(x)$ from $L_k(x)$ through

$$p_{m-1} = q_m - x_k p_m, \quad m = k, k-1, \dots, 2, 1, 0, \quad (\text{B.2})$$

where q_{-1} is the remainder when dividing $L_k(x)$ by $(x + x_k)$ which is zero since $(x + x_k)$ is a factor of $L_k(x)$. Hence, the recursion can be computed only for $m = k, k-1, \dots, 2, 1$.

REFERENCES

- [1] L. Aceto and D. Trigiante, *On the A-stable methods in the GBDF class*, Nonlinear analysis: real-world applications, *3* (2002), 9–23.
- [2] R. L. Bagley and P. J. Torvik, *A theoretical basis for the application of fractional calculus to viscoelasticity*, Journal of Rheology, *27* (1983), 201–210.
- [3] E. Barkai, R. Metzler, and J. Klafter, *From continuous time random walks to the fractional Fokker-Planck equation*, Physical Review E, *61* (2000), 132.
- [4] W. G. Bickley, *Formulae for numerical differentiation*, The Mathematical Gazette, *25* (1941), 19–27.
- [5] L. Eulero, *De Progressionibus Transcendentibus, seu quarum Termini Generales Algebraice Dari Nequevent Translation: “On Transcendental Progressions, That is, Those Whose General Terms Cannot be Given Algebraically”*, Commentarii academiae scientiarum Petropolitanae, *5* (1738), 36–57.
- [6] B. Fornberg, *Generation of finite difference formulas on arbitrarily spaced grids*, Mathematics of computation, *51* (1988), 699–706.
- [7] B. Fornberg, *Classroom note: Calculation of weights in finite difference formulas*, SIAM review, *40* (1998), 685–691.
- [8] L. Galeone and R. Garrappa, *On multistep methods for differential equations of fractional order*, Mediterranean Journal of Mathematics, *3* (2006), 565–580.
- [9] W. A. Gunarathna, H. M. Nasir, and W. B. Daundasekera, *An explicit form for higher order approximations of fractional derivatives*, Applied Numerical Mathematics, *143* (2019), 51–60.
- [10] A. H. Hadian Rasanan, N. Bajalan, K. Parand, and J. A. Rad, *Simulation of nonlinear fractional dynamics arising in the modeling of cognitive decision making using a new fractional neural network*, Mathematical Methods in the Applied Sciences, *43* (2020), 1437–1466.
- [11] H. Z. Hassan, A. A. Mohamad, and G. E. Atteia, *An algorithm for the finite difference approximation of derivatives with arbitrary degree and order of accuracy*, Journal of Computational and Applied Mathematics, *236* (2012), 2622–2631.
- [12] H. B. Keller and V. Pereyra, *Symbolic generation of finite difference formulas*, Mathematics of Computation, *32* (1978), 955–971.
- [13] I. R. Khan and R. Ohba, *Taylor series based finite difference approximations of higher-degree derivatives*, Journal of Computational and Applied Mathematics, *154* (2003), 115–124.
- [14] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham. P. Pir, L. N. Soldatova, et al., *The automation of science*, Science, *324* (2009), 85–89.
- [15] G. W. Leibnitz, *Letter from Hanover, Germany to G.F.A. L’Hospital, September 30, 1695*, Leibnizen Mathematische Schriften, Reprinted in Hildesheim, Germany (Olms Verlag), Olms Verlag Hildesheim, Germany, *2* (1962), 301–302.
- [16] S. K. Lele, *Compact finite difference schemes with spectral-like resolution*, Journal of computational physics, *103* (1992), 16–42.
- [17] C. Lubich, *Discretized fractional calculus*, SIAM Journal on Mathematical Analysis, *17* (1986), 704–719.
- [18] F. Mainardi, *Fractional relaxation-oscillation and fractional diffusion-wave phenomena*, Chaos, Solitons & Fractals, *7* (1996), 1461–1477.
- [19] M. M. Meerschaert and C. Tadjeran, *Finite difference approximations for fractional advection–dispersion flow equations*, Journal of Computational and Applied Mathematics, *172* (2004), 65–77.
- [20] R. Metzler and J. Klafter, *The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics*, Journal of Physics A: Mathematical and General, *37* (2004), R161.



- [21] H. M. Nasir and K. Nafa, *Algebraic construction of a third order difference approximation for fractional derivatives and applications*, ANZIAM Journal, 59 (2018), C231–C245.
- [22] H. M. Nasir and K. Nafa, *A new second order approximation for fractional derivatives with applications*, SQU Journal of Science, 23 (2018), 43–55.
- [23] I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Academic Press, 1998.
- [24] L. B. Rall and R. LB, *Automatic differentiation: Techniques and applications*, (1981).
- [25] B. Ross, *The development of fractional calculus 1695–1900*, Historia Mathematica, 4 (1977), 75–89.
- [26] B. Sadiq and D. Viswanath, *Finite difference weights, spectral differentiation, and superconvergence*, Mathematics of Computation, 83 (2014), 2403–2427.
- [27] C. Taylor, *Finite Difference Coefficients Calculator*, <http://web.media.mit.edu/~crtaylor/calculator.html>, (2017), Accessed: 2021-05-22.
- [28] B. M. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu, *Some approximations of fractional order operators used in control theory and applications*, Fractional calculus and applied analysis, 3 (2000), 231–248.
- [29] M. Weilbeer, *Efficient numerical methods for fractional differential equations and their analytical background*, Papierflieger Clausthal-Zellerfeld, Germany, 2006.
- [30] Y. Zhang, J. Gao, J. Peng, and W. Han, *A robust method of computing finite difference coefficients based on Vandermonde matrix*, Journal of Applied Geophysics, 152 (2018), 110–117.

