MATEMÁTICA
ESPOL – FCNM JOURNAL

2026, Vol. 24, No.2 Julio - Diciembre

https://www.revistas.espol.edu.ec/index.php/matematica/issue/archive

A study on the use of conformable fractional derivatives in the Newton–Raphson Method

Esther D. Gutiérrez M.*, Freddy R. Muñoz S., Christian F. Calderón G. and Luis F. Mejías

Recepción: 6/4/2026 **Aceptación: 9/7/2026** **Publicación: 28/7/2026**

Abstract This paper investigates the application of conformable fractional derivatives to iterative methods for finding the roots of nonlinear equations. We review the definition of the conformable fractional derivative and its generalization. Motivated by this framework, we introduce two perturbation functions, $e^{(\alpha-1)x}$ and $e^{(1-\alpha)x}$, with $\alpha \in (0,1]$, and incorporate them into a Newton–Raphson-type scheme. A theoretical analysis establishing local linear convergence is presented, together with numerical experiments implemented in Python. The proposed approach is evaluated on quadratic and cubic polynomials by comparing the two perturbation functions. The results indicate that, for suitable values of α, the proposed method may converge in fewer iterations than the classical Newton–Raphson method and can even converge in situations where the classical method fails.

Keywords fractional iterative schemes · root-finding algorithms · conformable calculus · polynomial root approximation · numerical convergence

2020 Mathematics Subject Classification 26A33 · 65H05 · 65H10 · 65D99

Corresponding Author*: Esther D. Gutiérrez
Departamento de Física, Facultad de Ciencias Naturales y Matemáticas, ESPOL, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador., e-mail: egutierr@espol.edu.ec

Freddy Roberto Muñoz Solis
Departamento de Matemática, Facultad de Ciencias Naturales y Matemáticas, ESPOL, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador., e-mail: frmunoz@espol.edu.ec

Christian Fernando Calderón González
Departamento de Matemática, Facultad de Ciencias Naturales y Matemáticas, ESPOL, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador., e-mail: chrfcald@espol.edu.ec

Luis Fernando Mejías
Departamento de Matemática, Facultad de Ciencias Naturales y Matemáticas, ESPOL, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador., e-mail: lfmejias@espol.edu.ec

1.1 Introduction

The Newton–Raphson method is one of the classical iterative procedures for approximating zeros of nonlinear equations [2, 10]. Its local convergence properties and its dependence on the derivative of the function make it a natural reference point for developing and comparing modified root-finding schemes. In particular, the central role played by the derivative raises the question of how this method may be affected when alternative notions of differentiation are considered.

Some attempts have been made to extend the concept of derivative so that its order is not necessarily an integer [7]. In these formulations, a fractional derivative is defined and then compared with the classical derivative through its mathematical properties and applications.

Among the numerous formulations proposed in the literature, the most widely studied are the Riemann–Liouville and Caputo derivatives [7], which have been extensively used in mathematical and physical modeling.

Diethelm [7] points out that, although the first developments of fractional calculus date back to the nineteenth century, the field has experienced remarkable growth during the last few decades. In recent years, several approaches have been proposed to define fractional derivatives, leading to new applications in engineering, physics, economics, and other areas of the applied sciences.

An example from economics is provided by Batrancea *et al.* [3], who compute Fisher information using Caputo fractional derivatives. According to the authors, this methodology captures long-range memory effects and highlights the importance of incorporating memory into information-theoretical models for a better understanding of the relationships among financial indicators.

Hernandez-Gomez *et al.* [8] developed a framework based on conformable fractional derivatives and applied it to a Gompertz model for tuberculosis. Their results show that, for suitable values of α, the fractional model provides a better fit to the available data than its classical counterpart.

Beyond their applications in modeling, fractional derivatives have also been incorporated into numerical algorithms. Candelario [1] works within a general framework involving Riemann–Liouville, Caputo, and conformable fractional derivatives to develop fractional versions of classical iterative methods. In particular, the author analyzes the stability and convergence of these fractional schemes and compares their order of convergence with that of their classical counterparts. These results suggest that fractional derivatives can be used not only to model complex phenomena, but also to construct modified iterative methods for solving nonlinear equations.

Motivated by this perspective, we consider the perturbation functions $T_1(\alpha,x) = e^{(1-\alpha)x}$ and $T_2(\alpha,x) = e^{(\alpha-1)x}$ within the framework of generalized conformable derivatives. Using these functions, we propose a Newton–Raphson-type iterative scheme obtained by replacing the classical derivative with the corresponding generalized conformable derivative.

An additional advantage of the proposed derivative is that its evaluation reduces to the computation of a classical derivative multiplied by a perturbation function, which considerably simplifies its numerical implementation.

The numerical experiments reveal the existence of threshold values of the parameter α beyond which the proposed conformable Newton–Raphson method converges for both perturbation functions. Furthermore, for certain ranges of α, the proposed scheme converges in fewer iterations than the classical Newton–Raphson method and may even converge in situations where the classical method fails.

The remainder of this paper is organized as follows. Section 1.2 presents the basic concepts of fractional calculus, the generalized conformable derivative, and the notation used throughout the manuscript. Section 1.3 introduces the conformable Newton–Raphson scheme and establishes its local convergence properties. Section 1.4 describes the numerical implementation and presents the computational experiments carried out for the proposed perturbation functions. Finally, the main conclusions and future research directions are discussed.

1.2 Preliminares

1.2.1 Fractional Derivatives

In this section we introduce a few different notions of fractional derivatives and establish the one we work with. As usual, we denote the gamma function by Γ and by $L_1[a,b]$ the space of the Lebesgue-integrable functions on $[a,b]$.

We take the next definitions from [7].

Definition 1. Let $\alpha > 0$, $a,x \in \mathbb{R}$ and $f \in L_1[a,b]$, the *Riemman-Louville fractional derivative* is defined by:

$$D_a^\alpha f(x) = \begin{cases} \dfrac{1}{\Gamma(m-\alpha)} \dfrac{d^m}{dx^m} \displaystyle\int_a^x \dfrac{f(t)}{(x-t)^{\alpha+1-m}}\, dt, & m-1 < \alpha < m \in \mathbb{N}, \\ \dfrac{d^m f(x)}{dx^m}, & \alpha \in \mathbb{N}. \end{cases}$$

Some other approach was developed to meet the fractional operator applied to a constant is 0.

Definition 2. Let $f : [a,b] \to \mathbb{R}$ be such that $\frac{d^m f(x)}{dt^m} \in L_1[a,b]$ with $m-1 < \alpha < m$. Then the *Caputo fractional derivative* is defined by:

$$\hat{D}_a^\alpha f(x) = \begin{cases} \dfrac{1}{\Gamma(m-\alpha)} \displaystyle\int_a^x (x-t)^{m-\alpha-1} \dfrac{d^m}{dt^m} f(t)\, dt, & m-1 < \alpha < m \in \mathbb{N}. \\ \dfrac{d^m f(x)}{dx^m}, & \alpha \in \mathbb{N}. \end{cases} \tag{1}$$

In the field of differential equations, fractional calculus has been developed using different approaches to fractional derivatives. It covers questions analogous to the existence and uniqueness of solutions, with a particular focus on initial value problems (Cauchy problems). Diethelm [7] studies equations involving Riemann–Liouville and Caputo fractional operators and develops fractional analogues of classical results for ordinary differential equations. In this context, Diethelm *et al.* [6] discuss an Adams-type predictor–corrector method for the numerical solution of equations involving the Caputo operator, $G^{\alpha}y(x) = f(x, y(x))$. This method has become one of the standard approaches for simulating fractional dynamical systems.

Moreover, Riemann–Liouville and Caputo operators have been widely used to model memory effects in dynamical systems. Caputo [5] and Bagley and Torvik [4], for example, showed that fractional derivatives can describe dissipation and viscoelastic behavior using significantly fewer parameters than classical integer-order models.

Another approach to fractional derivatives was introduced by Khalil *et al*. [9] in the following way.

Definition 3. Let $f : [0,\infty) \to \mathbb{R}$ be a function and $\alpha \in (0,1)$. We say that f is α-*differentiable* at $x > 0$ if the following limit exists:

$$\lim_{h\to 0} \frac{f(x+hx^{1-\alpha}) - f(x)}{h}. \tag{2}$$

In that case, the value of the limit is called the *conformable fractional derivative of order* α of f at x and it is denoted by. We say f is α-*differentiable* at 0 if there is an $a > 0$ such that $T_\alpha f(x)$ exists for all $x \in (0,a)$ and the limit

$$\lim_{x\to 0+} T_\alpha f(x) \tag{3}$$

exists, in this case its value is denoted by $T_\alpha f(0)$.

Unlike integral-based definitions, the conformable derivative preserves fundamental properties of standard calculus, as the following propositions show (for the proofs see [9].)

Theorem 1. *If f is α-differentiable at x then f is continuous at x.*

Theorem 2. *If f and g are α-differentiable at x then fg is α-differentiable at x and*

$$T_\alpha(fg)(x) = f(x)T_\alpha g(x) + g(x)T_\alpha f(x). \tag{4}$$

These ideas were set in a more general framework in the work of Hernández-Gómez *et al.* [8]. The starting point is the following definition.

Definition 4. Let $I \subset \mathbb{R}$ be an open interval and $\alpha > 0$. If f and $T(\alpha,\cdot)$ are two functions defined on I such that $T(\alpha,\cdot)$ is continuous and positive, We say that f is G^{α}_{T}-*differentiable* at $x \in I$ if the following limit exists:

$$\lim_{h\to 0} \frac{1}{h^{\lceil\alpha\rceil}} \sum_{k=0}^{\lceil\alpha\rceil} (-1)^k \binom{\lceil n\rceil}{k} f(x - khT(\alpha,x)). \tag{5}$$

In that case, the value of the limit is called the *general conformable fractional derivative of order* α of f at x and it is denoted by $G_T^\alpha f(x)$. If $a = \inf I$ or $b = \sup I < +\infty$ we define

$$G_T^\alpha f(a) = \lim_{h\to 0+} \frac{1}{h^{\lceil\alpha\rceil}} \sum_{k=0}^{\lceil\alpha\rceil} (-1)^k \binom{\lceil n\rceil}{k} f(a - khT(\alpha,a)), \tag{6}$$

$$G_T^\alpha f(b) = \lim_{h\to 0-} \frac{1}{h^{\lceil\alpha\rceil}} \sum_{k=0}^{\lceil\alpha\rceil} (-1)^k \binom{\lceil n\rceil}{k} f(b - khT(\alpha,b)). \tag{7}$$

The generalized conformable derivative satisfies several properties that resemble those of the classical derivative. In particular, for $\alpha \in (0,1)$, a function is G_T^α-differentiable if and only if it is differentiable in the classical sense. Moreover, several standard rules of differential calculus, including continuity and Leibniz-type properties, are preserved within this framework. Unlike the Riemann–Liouville and Caputo operators, these features make conformable derivatives particularly attractive for the construction of iterative numerical methods.

Hernandez-Gomez *et al.* [8] studied a particular case of the generalized conformable derivative obtained by choosing the perturbation function $T(\alpha,x) = e^{(\alpha-1)x}$. Using this operator, the authors developed a corresponding theory of fractional differential equations and applied it to the Gompertz population growth model

$$G_T^\alpha y(t) = r\,y(t) \ln\left(\frac{k}{y(t)}\right),$$

where r and k denote the intrinsic growth rate and carrying capacity, respectively. Their analysis showed that, for suitable values of α, the resulting model provided a better fit to the available data than models based on classical, Caputo, and Khalil derivatives. This work illustrates the flexibility of the generalized conformable framework and motivates the study of alternative perturbation functions.

From Definition 4, it follows that choosing $T(\alpha,x) = x^{1-\alpha}$ with $\alpha \in (0,1]$ recovers the conformable derivative introduced by Khalil *et al.* This observation highlights the flexibility of the generalized framework and motivates the consideration of alternative perturbation functions. In the present work, we focus on the perturbation functions T_1 and T_2 introduced in Section 1.1.

In this work, we propose a specific formulation of the conformable fractional derivative, which is a particular case of G_T^α as in Definition 4 when $T(\alpha,x) = e^{(1-\alpha)x}$. In this case we use f^α instead of G_T^α. Thus we have

$$f^\alpha(x) = \lim_{h\to 0} \frac{f(x) - f(x - he^{(1-\alpha)x})}{h}, \tag{8}$$

again defining the fractional derivative appropriately at the endpoints of the interval. In this case, we have the following result.

Theorem 3. *Suppose that $\alpha \in (0,1]$. The function $f : I \to \mathbb{R}$ is α-differentiable at $x \in I$ if and only if f is differentiable at x. In any case we have*

$$f^{\alpha}(x) = e^{(1-\alpha)x} f'(x). \tag{9}$$

Proof. Equation (9) is obtained following both directions in the following sequence of equalities, with the substitution $\mu = -e^{(1-\alpha)x}h$: we have that $\mu \to 0$ as $h \to 0$, we have

$$\begin{aligned}
f^{\alpha}(x) &= \lim_{h\to 0} \frac{f(x) - f(x - he^{(1-\alpha)x})}{h} \\
&= \lim_{\mu\to 0} \frac{f(x) - f(x+\mu)}{-\mu e^{(\alpha-1)x}} \\
&= e^{(1-\alpha)x} \lim_{\mu\to 0} \frac{f(x+\mu) - f(x)}{\mu} \\
&= e^{(1-\alpha)x} f'(x).
\end{aligned}$$

The following definition is included for completeness and follows the spirit of generalized Taylor expansions associated with conformable derivatives.

Definition 5. Let $f : I \to \mathbb{R}$ continuously differentiable in a neighborhood of $x_0 \in I$ for $\alpha \in (0,1]$, we define the *fractional Taylor expansion* as:

$$\sum_{k=0}^{\infty} \frac{(f^{\alpha})^{(k)}(x_0)}{k!}(x - x_0)^k$$

for $x \in (x-h, x+h)$, $h > 0$, where $(f^{\alpha})^{(k)}$ means the fractional derivative applied k times.

1.2.2 Classical Newton–Raphson Method

Many iterative methods have been developed for solving nonlinear equations. Among them, the Newton–Raphson method is one of the most widely used because of its simplicity and its quadratic local convergence. Since the conformable scheme proposed in this work is obtained as a modification of the classical Newton–Raphson iteration, we briefly recall its formulation and convergence properties.

This method consists on constructing a sequence $\{x_n\}$ in the following way: If c is a zero of f, choose point x_1 close enough to c, then take x_2 as the intercept of the tangent line at $(x_1, f(x_1))$ and the x-axis, that is

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Similarly, consider

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

and so on. Then, under some conditions $\{x_n\}$ converges to c.

Formally, the method is established in the following way (for a proof see [10]).

The geometric construction above leads to the iterative scheme (10). Under suitable regularity assumptions, the following classical convergence result holds.

Theorem 4 (Newton–Raphson method). *Let $f : I \to \mathbb{R}$, a continuously differentiable function on the interval I that have a zero c. Suppose that $f'(c) \neq 0$. Let $x_1 \in I$ such that $|x_1 - c| < \delta$ for $\delta > 0$. Let $\{x_n\}$ be the sequence defined for $n > 1$ as:*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad \textit{for } n > 1. \tag{10}$$

Then $\{x_n\}$ converges to c and its convergence is quadratic.

A condition for convergence, which will be explained later, can be deduced from the following theorem, which we adapt to the context of metric spaces. For more information, consult [10].

Theorem 5 (Fixed Point Theorem). *Let (X,d) be a metric space. A mapping $T : X \to X$ is called a contraction if there exists a constant $L \in [0,1)$ such that, for every $x_1, x_2 \in X$,*

$$d(T(x_1), T(x_2)) \leq L \cdot d(x_1, x_2)$$

then,

- *There is a unique fix point $c \in X$ such that $T(c) = c$*
- *For every initial point $x_0 \in X$, the successive approximations $x_{n+1} = T(x_n)$ converge to the fixed point c*

1.3 Conformable Newton–Raphson method

The flexibility of conformable derivatives has motivated the development of fractional versions of classical iterative methods. In particular, Candelario [1] proposed a conformable Newton-type method based on a generalized conformable derivative and established convergence results for the resulting iteration.

If $f : [a, +\infty) \to \mathbb{R}$ with $\alpha \in (0,1]$, $a < x$ the derivative T^a_α of f is defined by:

$$T^a_\alpha f(x) = \lim_{h \to 0} \frac{f(x + h(x-a)^{1-\alpha}) - f(x)}{h}.$$

Appropriately defining the derivative at endpoints, if f is α-differentiable on (a,b) with $b \in \mathbb{R}$

$$T_\alpha^a f(a) = \lim_{x\to a^+} T_\alpha^a f(a).$$

Candelario [1] proposed a conformable variant of the Newton–Raphson method and showed that the resulting iteration exhibits at least quadratic convergence independently of the value of α. Numerical examples were also presented illustrating improvements with respect to the classical Newton method.

Motivated by the classical Newton–Raphson construction, we derive a conformable counterpart by replacing the classical derivative in the first-order approximation of the function with the conformable derivative introduced in Theorem 3.

Suppose that:

$$f(x_k) \approx f(x) + f^\alpha(x)(x_k - x).$$

Then we have,

$$x = x_k - \frac{f(x_k)}{f^\alpha(x)}$$

Taking x_k, x_{k+1} as approximations of x

$$x_{k+1} = x_k - \frac{f(x_k)}{f^\alpha(x_k)}$$

for $k \in \mathbb{N} \cup \{0\}$.

The following result establishes a lower theoretical bound for the convergence rate of the proposed method. Although only linear convergence is proved analytically, the numerical experiments reported in Section 1.4 reveal a substantially stronger behavior for the polynomial examples considered in this work.

For definiteness, the proof is presented for the perturbation function $T_2(x) = e^{(1-\alpha)x}$. The argument for $T_1(x) = e^{(\alpha-1)x}$ follows analogously.

Theorem 6 (Linear convergence of the conformable Newton–Raphson method). *Let $f \in C^2(I)$ and suppose that $c \in I$ is a simple zero of f, that is, $f(c) = 0$ and $f'(c) \neq 0$.*

Let $\{x_k\}$ be the sequence generated by

$$x_{k+1} = x_k - \frac{f(x_k)}{f^\alpha(x_k)}$$

Then the iteration converges locally to c with at least linear convergence.

Proof. Suppose c is a zero of f, expanding the first order Taylor polynomial about c for f and f' we have:

$$f(x_k) = f'(c)e_k + O(e_k^2),$$
$$f^\alpha(x_k) = e^{(1-\alpha)x_k}(f'(c) + f''(c)e_k + O(e_k^2).$$

Replacing the respective Taylor polynomial for $e^{(1-\alpha)x_k}$, we get

$$f^\alpha(x_k) = (e^{(1-\alpha)c} + (1-\alpha)e^{(1-\alpha)c}e_k + O(e_k^2)) \cdot (f'(c) + f''(c)e_k + O(e_k^2)).$$

Denote $T_2(x) = e^{(1-\alpha)x}$, so we can write:

$$f^{\alpha}(x_k) = T_2(c)f'(c) + \big(T_2(c)f''(c) + T_2'(c)f'(c)\big) \cdot e_k + O(e_k^2).$$

This yields

$$x_k - \frac{f(x_k)}{f^{\alpha}(x_k)} = x_k - \frac{f'(c)e_k + O(e_k^2)}{T_2(c)f'(c) + O(e_k)}.$$

Then, taking a factor $f'(c) \neq 0$ and denote $e_{k+1} = x_{k+1} - c$

$$e_{k+1} + c = e_k + c - \frac{e_k}{T_2(c) + O(e_k)} + O(e_k^2),$$
$$e_{k+1} \approx e_k - \frac{e_k}{T_2(c)} = \left(1 - \frac{1}{T_2(c)}\right) e_k = (1 - e^{(\alpha-1)c})e_k.$$

This proves the linear convergence.

The previous result can be complemented by a fixed-point argument based on the iteration function and treated formally in [2]. Defining

$$g(x) = x - \frac{f(x)}{f^{\alpha}(x)}$$

and applying the sufficient condition $|g'(x)| < 1$.

The result follows as: Taking into account theorem 3, we get that:

$$g'(x) = 1 - \frac{f'(x)}{e^{(1-\alpha)x}f'(x)} + f(x) \cdot \frac{d}{dx}\left(e^{(1-\alpha)x}f'(x)\right) \cdot \frac{1}{(e^{(1-\alpha)x}f'(x))^2}.$$

Evaluating g' at the zero x^* of f then we have:

$$|g'(x^*)| = |1 - e^{(\alpha-1)x^*}| < 1.$$

We can write this condition as

$$(1-\alpha)x^* > -\log 2. \tag{11}$$

The fixed-point analysis presented below provides an explicit sufficient condition guaranteeing the contraction property of the iteration. This provides a sufficient local condition for the convergence of the proposed method.

1.4 Examples and numerical experiments

In this section, we present numerical experiments comparing two variants of the conformable Newton–Raphson method associated with different perturbation functions.

We analyze three polynomial test problems, covering simple roots, multiple roots, and a case in which the classical Newton method fails to converge. The goal of these experiments is not only to validate the theoretical convergence analysis, but also to highlight qualitative differences between the two perturbative functions and to identify parameter regimes in which the conformable method outperforms the classical Newton–Raphson scheme.

1.4.1 Numerical setup

Unlike the Riemann–Liouville and Caputo fractional derivatives, whose numerical evaluation often requires discretization schemes such as the Grünwald–Letnikov approximation, the conformable derivative considered in this work admits the explicit expression

$$p_T^{\alpha}(x) = T(\alpha, x)p'(x).$$

Therefore, no numerical approximation of the fractional derivative is required. For polynomial functions, the derivative $p'(x)$ is computed analytically from the polynomial coefficients, and the conformable derivative is obtained directly from the previous expression.

Figure 1.1 summarizes the computational workflow adopted in all numerical experiments. For each polynomial and each value of α, the conformable derivative is evaluated, the iterative method is executed until convergence or until the maximum number of iterations is reached, and the resulting convergence statistics are recorded.

The numerical experiments are carried out in Python using double-precision arithmetic. For each test polynomial p, we fix the initial guess $x_0 = -1.0$ and consider a uniform grid of $N = 400$ values of α in the interval $[0.1, 1]$. For a given α, we apply the conformable Newton–Raphson scheme:

$$x_{n+1} = x_n - \frac{p(x_n)}{p^{\alpha}(x_n)}, \qquad n = 0, 1, \ldots, \tag{12}$$

where p^{α} denotes the conformable derivative associated with the selected perturbation function $T(\alpha, x)$.

In this work, we consider the following two specific choices:

$$T_1(\alpha, x) = e^{(\alpha-1)x}, \qquad p_{T_1}^{\alpha}(x) = e^{(\alpha-1)x}\, p'(x), \tag{13}$$

$$T_2(\alpha, x) = e^{(1-\alpha)x}, \qquad p_{T_2}^{\alpha}(x) = e^{(1-\alpha)x}\, p'(x). \tag{14}$$

The iterative scheme is stopped when $|x_{n+1} - x_n| < \text{tol\_}x$ which indicates convergence, or when the maximum number of iterations $n_{\max} = 50$ is reached, the method is declared not convergent.

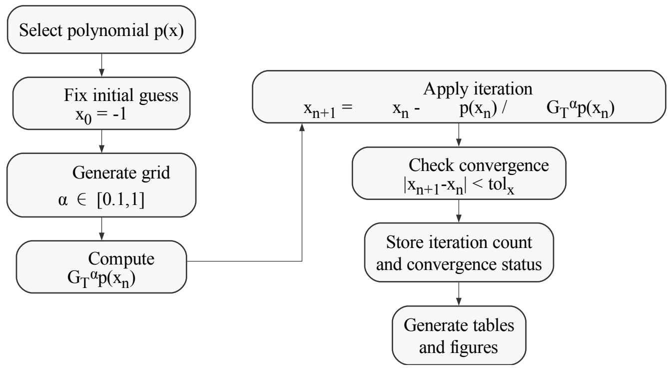

Fig. 1.1: Computational workflow used to perform the numerical experiments for the conformable Newton–Raphson method.

To avoid numerical instabilities, we declare a failure whenever $|p_T^\alpha(x_n)| < 10^{-14}$ or a non-finite value (NaN/Inf) occurs. For each α we record (i) whether convergence is achieved and (ii) the number of iterations required. In summary tables we report the subset of α values for which convergence occurs in at most 10 iterations, while scatter plots display the full dependence of the iteration count on α up to the cap $n_{\max}$.

1.4.2 Quadratic polynomial: $x^2 - 2$

Although the theoretical analysis guarantees only linear convergence, the numerical experiments indicate a robust convergence behavior for polynomial functions. As a first benchmark, we consider the classical quadratic polynomial $p(x) = x^2 - 2$, which serves as a standard reference example for Newton–type methods, to investigate how the fractional parameter α influences the convergence of the conformable Newton–Raphson method.

We evaluate the convergence across the spectrum $\alpha \in (0,1]$, employing the two perturbation choices introduced in Section 1.4. Specifically, we consider the two cases corresponding to T_1 and T_2.

Table 1 highlights the values of α for which each variant converges in fewer than 10 iterations, while Figure 1.2 shows the global dependence of the iteration count on α.

Table 1: Convergence results for α values where $p(x) = x^2 - 2$.
Parameters: $x_0 = -1.0$, tol_$x = 10^{-9}$, $n_{\max} = 50$.

	Case 1		**Case 2**	
α	**Conv** ≤ 10	**Iterations**	**Conv** ≤ 10	**Iterations**
0.303008	False	40	True	4
0.305263	False	40	True	4
0.327820	False	39	True	5
0.330075	False	39	True	5
0.332331	False	38	True	5
0.334586	False	38	True	5
0.336842	False	38	True	5
0.339098	False	38	True	5
0.341353	False	38	True	5
0.343609	False	38	True	5
0.345865	False	37	True	5
0.451880	False	31	True	7
0.454135	False	31	True	7
0.456391	False	31	True	7
0.492481	False	29	True	9
0.918797	True	10	False	11
0.921053	True	10	False	11
0.923308	True	10	False	11
0.925564	True	10	False	11
0.927820	True	10	False	11
		...		
0.993233	True	7	True	7
0.995489	True	6	True	6
0.997744	True	6	True	6
1.000000	True	5	True	5

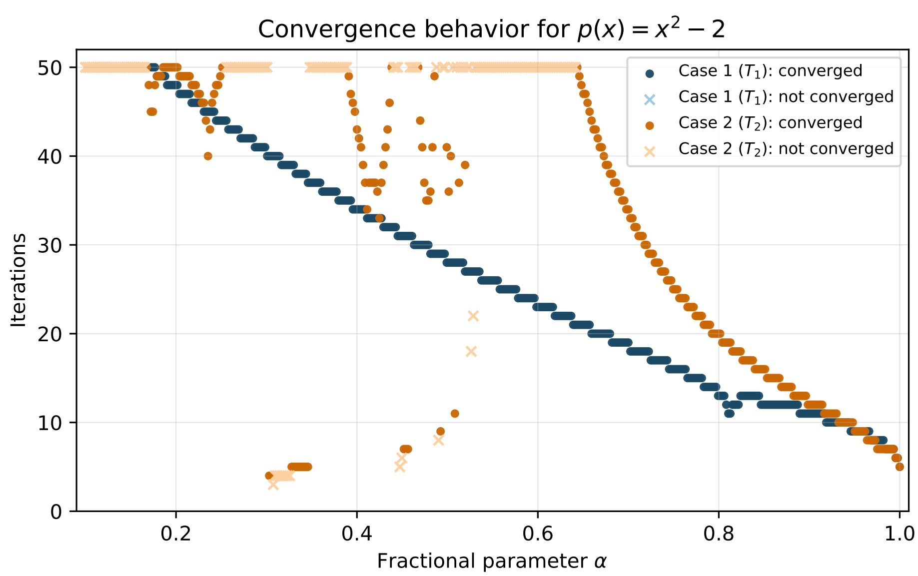

Fig. 1.2: Convergence behavior of the conformable method for Case 1 (T_1) and Case 2 (T_2) applied to $p(x) = x^2 - 2$.

The results suggest the existence of a threshold value of α beyond which both cases of the conformable Newton–Raphson method consistently converge within the prescribed iteration limit. Moreover, Case 2 exhibits a wider range of α values with non-monotonic convergence behavior, including isolated regions where convergence is achieved in fewer iterations, such as around $\alpha \approx 0.303$.

1.4.3 Cubic polynomial with Newton–Raphson method's failure: $x^3 - 2$

To further assess the performance of the method, we consider a cubic polynomial for which the classical Newton–Raphson method fails to converge from the chosen initial condition.

Table 2: Convergence results for α values where $p(x) = x^3 - 2$.
Parameters: $x_0 = -1.0$, tol_x, $n_{\max} = 50$.

	Case 1		**Case 2**	
α	**Conv** ≤ 10	**Iterations**	**Conv** ≤ 10	**Iterations**
0.185714	True	7.0	False	33.0
0.206015	True	7.0	False	39.0
0.208271	True	7.0	False	39.0
0.210526	True	7.0	False	39.0
0.212782	True	7.0	False	39.0
0.233083	True	9.0	False	39.0
0.239850	True	9.0	False	38.0
0.280451	True	10.0	False	37.0
0.305263	True	8.0	False	35.0
0.307519	True	8.0	False	35.0
0.366165	True	6.0	False	32.0
0.368421	True	6.0	False	32.0
0.370677	True	6.0	False	31.0
0.372932	True	6.0	False	31.0
0.375188	True	6.0	False	31.0
0.377444	True	6.0	False	31.0
0.422556	True	6.0	False	29.0
0.424812	True	6.0	False	30.0
0.449624	True	7.0	False	30.0
0.460902	True	7.0	False	30.0
		...		
0.990977	False	NaN	True	3.0
0.993233	False	NaN	True	3.0
0.995489	False	NaN	True	3.0
0.997744	False	NaN	True	3.0
1.0	NaN	NaN	NaN	NaN

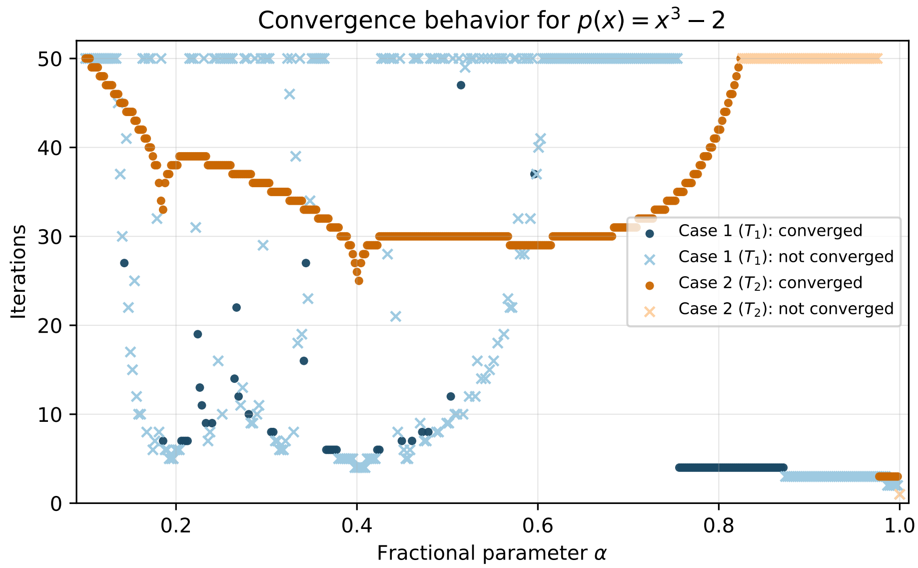

Fig. 1.3: Convergence behavior of the conformable method for Case 1 (T_1) and Case 2 (T_2) applied to $x^3 - 2$

As in the example $x^2 - 2$ we show the respective Table 2 and Figure 1.3. We observe for this implementation for $\alpha = 1$ fails the convergence but the two variations 1 and 2 of the conformable Newton–Raphson method converge in fewer than 10 iterations. Another fact is that Case 2 again converge in fewer iterations.

1.4.4 Cubic Polynomial with Multiple Roots

As final example we prove the conformable Newton–Raphson method in a polynomial with multiple roots, we chose $x^3 - 6x^2 + 11x - 6$. The setup of this implementation was the same as all the examples before, with the initial point $x_0 = -1.0$ the method converges to the real root $x = 1$.

Table 3: Convergence results for α values where $p(x) = x^3 - 6x^2 + 11x - 6$. Parameters: $x_0 = -1.0$, tol_$x = 10^{-9}$, $n_{\max} = 50$.

	Case 1		**Case 2**	
α	**Conv** ≤ 10	**Iterations**	**Conv** ≤ 10	**Iterations**
0.952632	True	10	False	11
0.954887	True	10	False	11
0.957143	True	10	False	11
0.959398	True	10	False	11
0.961654	True	10	False	11
0.963910	True	10	False	11
0.966165	True	10	False	11
0.968421	True	10	False	11
0.970677	True	10	False	11
0.972932	True	10	False	11
0.975188	True	10	True	10
0.977444	True	10	True	10
0.979699	True	10	True	10
0.981955	True	10	True	10
0.984211	True	10	True	10
0.986466	True	10	True	10
0.988722	True	10	True	10
0.990977	True	9	True	10
0.993233	True	9	True	9
0.995489	True	9	True	9
0.997744	True	8	True	9
1.000000	True	8	True	8

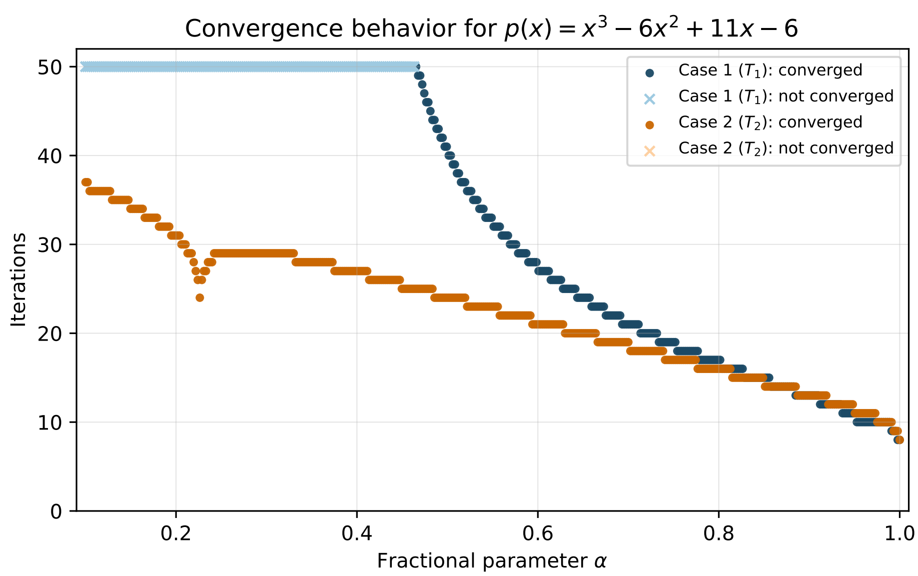

Fig. 1.4: Convergence behavior of the conformable method for Case 1 (T_1) and Case 2 (T_2) applied to $x^3 - 6x^2 + 11x - 6$

We present Table 3 and Figure 1.4 from which we conclude that although neither of the two variations of the method obtains convergence in fewer iterations comparing with $\alpha = 1$, there is a clear pattern of convergence of the two variations of the method. Furthermore, consistent with theoretical expectations, as $\alpha \to 1$, the convergence dynamics of both variations 1 and 2 align asymptotically with the behavior of the classical Newton–Raphson method.

1.4.5 Comparison between two variations

In all the examples presented raised an important question which of the two variants performs better?. Analyzing Figures 1.2, 1.3 and 1.4 that shows the global behavior, we can conclude the Case 2 show convergence in the majority of values of α for the three examples although in 1.2 the pattern of convergence become stable exceeding a specific threshold of α, this is the method that get the convergence in fewer iterations. Another conclusion is the convergence behavior of the variations of the method approaches to the classical method except in Figure 1.3 where for some values of α closest to 1 the method obtains convergence, which appears to be specific to this case.

1.4.6 Geometric Visualization of Convergence

To provide geometric insight into the convergence mechanism, we illustrate the iterative trajectory for the polynomial $p(x) = x^3 - 6x^2 + 11x - 6$. This specific example is illustrative because it allows us to visualize how the conformal derivative modifies the tangent slope to guide the search direction effectively. We selected a representative fractional order of $\alpha = 0.993233$, which was empirically determined to achieve convergence to the root $x = 1$ in exactly 9 iterations, starting from the initial guess $x_0 = -1.0$.

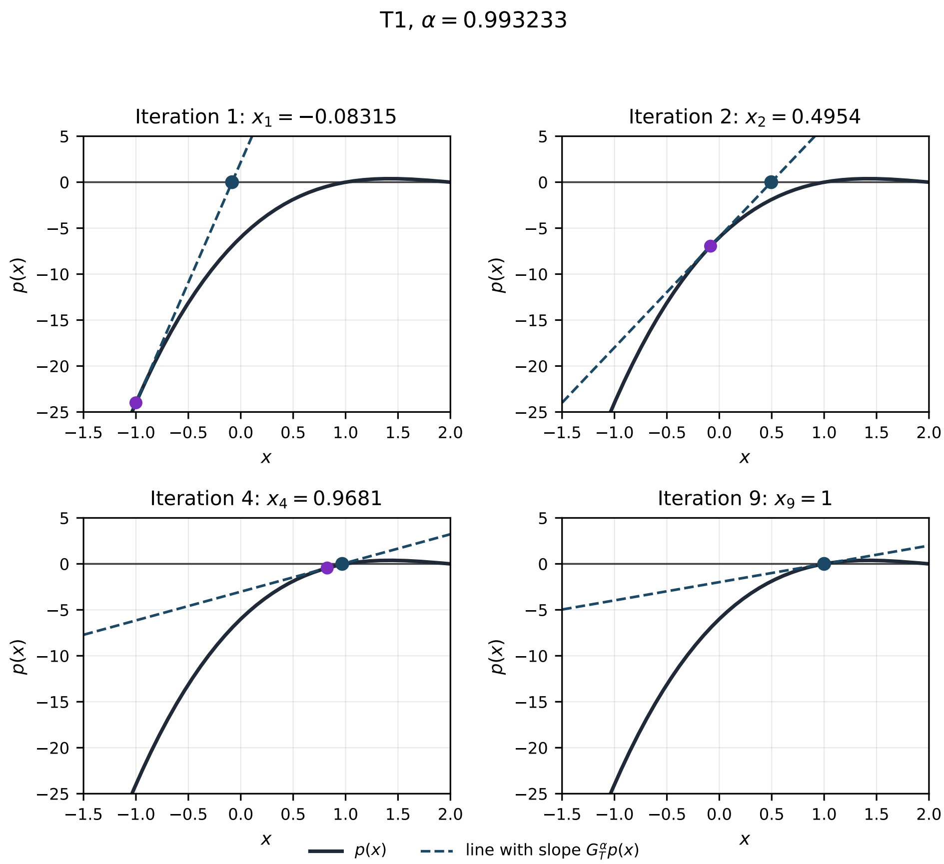

Fig. 1.5: Iterative trajectories of the conformable Newton–Raphson method applied to $p(x) = x^3 - 6x^2 + 11x - 6$, illustrating convergence for $\alpha = 0.993233$.

Table 4 details the numerical evolution of this sequence. It is worth noting the rapid reduction of the residual $p(x_k)$ in the final steps, characteristic of the method's efficiency near the root.

Table 4: Iteration history for $p(x) = x^3 - 6x^2 + 11x - 6$ with $\alpha = 0.993233$.

Iteration (k)	x_k	$p(x_k)$
0	-1.0000000000000000	-2.400e+01
1	-0.0831484511516762	-6.957e+00
2	0.4953566585091879	-1.902e+00
3	0.8248162330977182	-4.478e-01
4	0.9680850467097708	-6.692e-02
5	0.9987783866854815	-2.448e-03
6	1.0000060372397266	1.207e-05
7	0.9999999958952164	-8.210e-08
8	1.0000000000278710	5.574e-10
9	0.9999999999981072	-3.785e-12

Conclusions

This work investigated the use of generalized conformable derivatives in the construction of a Newton–Raphson-type iterative method for approximating roots of nonlinear equations. A theoretical analysis established local linear convergence of the proposed scheme, while numerical experiments implemented in Python demonstrated that the method can exhibit a substantially stronger performance for particular values of the parameter α.

The numerical results revealed the existence of threshold values of α beyond which convergence is consistently achieved for the perturbation functions considered. In addition, the proposed conformable scheme was able to converge in some situations where the classical Newton–Raphson method failed and, in several cases, required fewer iterations to reach the prescribed tolerance. Among the two perturbation functions studied, the formulation based on $e^{(1-\alpha)x}$ exhibited the most robust convergence behavior over the range of parameters considered.

An important advantage of the proposed approach is that the conformable derivative admits an explicit representation in terms of the classical derivative. Consequently, unlike other fractional operators, its implementation does not require the numerical approximation of a fractional derivative, making the method computationally simple and easy to extend.

The present study is subject to some limitations. The numerical analysis was restricted to polynomial functions and a fixed choice of initial conditions. Therefore, the results cannot be directly generalized to broader classes of nonlinear equations without further investigation.

Future work may focus on establishing stronger convergence results, studying the asymptotic behavior of the method for different perturbation functions, extending the analysis to transcendental functions and nonlinear systems, and developing conformable versions of other classical iterative schemes. Another promising direction is the design of adaptive strategies capable of selecting optimal values of α according to the characteristics of the problem under consideration.

References

1. G. G. Candelario Villalona, *Métodos iterativos fraccionarios para la resolución de ecuaciones y sistemas no lineales: diseño, análisis y estabilidad*, Tesis Doctoral, Departamento de Matemática Aplicada, Universitat Politècnica de València, Valencia, 2023.
2. R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed., Brooks/Cole, Cengage Learning, Boston, 2011.
3. L. M. Batrancea, Ö. Akgüller, M. A. Balcı, D. A. Koç, L. Gaban, *Memory-Driven Dynamics: A Fractional Fisher Information Approach to Economic Interdependencies*, Entropy, 27, 2025.
4. R. L. Bagley, P. J. Torvik, *A theoretical basis for the application of fractional calculus to viscoelasticity*, Journal of Rheology, 27(3), 201–210, 1983.
5. M. Caputo, *Linear models of dissipation whose Q is almost frequency independent–II*, Geophysical Journal International, 13(5), 529–539, 1967.
6. K. Diethelm, N. J. Ford, A. D. Freed, *A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dynamics, 29, 3–22, 2002.
7. K. Diethelm, *The Analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type*, Springer, Berlin, 2010.
8. J. C. Hernández-Gómez, R. Reyes, J. M. Rodríguez, and J. M. Sigarreta, Fractional model for the study of the tuberculosis in Mexico, *Mathematical Methods in the Applied Sciences*, **45**, 17, pp. 10675–10688, 2022.
9. R. Khalil, M. Al Horani, A. Yousef, and M. Sababheh, *A new definition of fractional derivative*, Journal of Computational and Applied Mathematics, vol. 264, pp. 65–70, 2014, doi:10.1016/j.cam.2014.01.002.
10. R. Kress, *Numerical analysis*, Graduate Texts in Mathematics, Vol. 181, Springer-Verlag, New York, 1998.
11. M. Spivak, *Calculus* (fourth edition), Publish or Perish, Houston, 2008.

Appendix: Python implementation and computational workflow

This appendix summarizes the Python implementation used to generate the numerical experiments reported in Section 1.4. The code was organized as a small reusable library so that the conformable Newton–Raphson method can be applied to different nonlinear functions and to different choices of the perturbation function $T(\alpha,x)$.

The objective of this appendix is not to provide the complete source code, but rather to document the main routines required to reproduce and extend the numerical experiments reported in this work.

It is important to remark that, for the conformable derivative considered in this work, no finite-difference approximation of the fractional derivative is required. Indeed, for differentiable functions one has

$$p_T^{\alpha}(x) = T(\alpha,x)p'(x),$$

so the implementation evaluates the conformable derivative from the classical derivative $p'(x)$ and the selected perturbation function. In the polynomial examples, $p'(x)$ is computed analytically from the polynomial coefficients.

The core routines of the Python library are shown below. The cases T_1 and T_2 correspond, respectively, to $T_1(\alpha,x) = e^{(\alpha-1)x}$ and $T_2(\alpha,x) = e^{(1-\alpha)x}$.

```
import numpy as np

def conformable_derivative_value(x, alpha,
derivative, case="T1"):
    if case.upper() in {"T1", "CASE1", "1"}:
        factor = np.exp((alpha - 1.0) * x)
    elif case.upper() in {"T2", "CASE2", "2"}:
        factor = np.exp((1.0 - alpha) * x)
    else:
        raise ValueError("case must be 'T1' or 'T2'.")
    return factor * derivative(x)


def conformable_newton(function, derivative,
x0, alpha, case="T1",
tol_x=1e-9, max_iter=50, eps_denom=1e-14):
    x = float(x0)
    history = [x]
    residual_history = [function(x)]

    for n in range(1, max_iter + 1):
        gx =
        conformable_derivative_value(x, alpha,
        derivative, case)
        if (not np.isfinite(gx)) or abs(gx) < eps_denom:
            return history, residual_history, False, n - 1

        x_new = x - function(x) / gx
        if not np.isfinite(x_new):
            return history, residual_history, False, n

        history.append(x_new)
        residual_history.append(function(x_new))

        if abs(x_new - x) < tol_x:
            return history, residual_history, True, n

        x = x_new

    return history, residual_history, False, max_iter
```

The library can also be used with any polynomial supplied through its coefficients. For example, if

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

then the coefficients are provided in decreasing degree order as $[a_n, a_{n-1}, \ldots, a_1, a_0]$. The following auxiliary routine constructs both the polynomial and its classical derivative.

```
def polynomial_from_coefficients(coefficients):
    p = np.poly1d(coefficients)
    dp = np.polyder(p)
    return p, dp
```

Thus, to apply the method to a polynomial different from those considered in the article, the user only needs to change the coefficient list, the initial condition, the value of α, and the perturbation case. For example, the method can be applied to any polynomial by modifying only the coefficient vector, the initial condition, the value of α, and the perturbation function. For the polynomial

$$p(x) = x^4 - 3x + 1$$

can be tested as follows.

```
coefficients = [1, 0, 0, -3, 1]
p, dp = polynomial_from_coefficients(coefficients)

history, residuals, converged, iterations =
conformable_newton(
    function=p,
    derivative=dp,
    x0=-1.0,
    alpha=0.8,
    case="T1",
    tol_x=1e-9,
    max_iter=50
)

print(converged, iterations)
print(history[-1], residuals[-1])
```

In this way, the same implementation can be extended to other polynomial equations without modifying the core routines.

The complete version of the library also includes routines for parameter sweeps in α, automatic generation of convergence tables, and production of publication-quality figures. The same computational framework was used for all numerical experiments reported in this manuscript.