OPEN ACCESS
Fractals (2025) 2540268 (14 pages)

DOI: 10.1142/S0218348X25402686

FRACTAL ANALYSIS OF FRACTIONAL NUMERICAL SCHEME FOR RESOLVING NONLINEAR ENGINEERING PROBLEMS

MUDASSIR SHAMS and NASREEN KAUSAR
Department of Mathematics, Faculty of Arts and Science, Balikesir University, Balikesir 10145, Turkey

HOSSEIN JAFARI
Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Chennai 602105, Tamil Nadu, India

Department of Mathematical Sciences, University of South Africa, UNISA0003, South Africa

Department of Medical Research, China Medical University Hospital China Medical University, Taichung 110122, Taiwan

GEORGIA IRINA OROS*
Department of Mathematics and Computer Science, University of Oradea, Oradea 410087, Romania
georgia_oros_ro@yahoo.co.uk

Received August 26, 2024
Accepted March 26, 2025
Published October 4, 2025

Abstract

Fractional calculus has evolved as an effective mathematical tool for simulating complex dynamical systems in science and engineering. This study develops an enhanced fractional iterative technique for solving nonlinear equations that incorporates fractional calculus to improve accuracy, stability and computational efficiency. Traditional approaches frequently struggle with

*Corresponding author.

high computing costs and slow convergence, but the suggested method effectively balances accuracy and efficiency by adding Caputo fractional-order derivatives. Analysis of convergence reveals that the order of convergence of the suggested family of approaches is $2ß + 1$. Fractal analysis of the suggested numerical methods for solving nonlinear equations reveals that they outperform existing classical approaches in terms of convergence and stability. A comparison with existing methods shows that the newly developed schemes perform better in terms of residual error reduction, CPU time and convergence rate. Since increasing the fractional parameter ß from 0 to 1 greatly increases the method's efficiency — near-optimal performance is seen around $ß \approx 1$ — it plays a critical role. The novel strategy generates fractals with greater elapsed time and consistency than existing strategies, according to numerical studies on engineering applications.

Keywords: Caputo Derivative; Theoretical Convergence; Computational Analysis; Elapsed Time; Percentage Convergence.

1. INTRODUCTION

Leibniz and L'Hopital proposed the semi-derivative[1–3] about 1695, which gave rise to both classical and fractional calculus. Fractional calculus, a field of mathematical analysis that extends classical calculus to non-integer orders, is gaining popularity because of its effectiveness in modeling complex systems with non-local and memory-dependent behaviors. Fractional derivatives — most notably, the Caputo derivative, which extends the concept of differentiation to include fractional orders — are the basis for fractional calculus theory. Caputo derivatives, unlike integer-order derivatives, capture the influence of past history on a system's current state, making them well suited for characterizing processes characterized by memory effects and long-range interactions. The Caputo derivative retains the advantages of classical calculus while allowing for the development of more comprehensive models capable of capturing the intricate dynamics seen in a wide range of real-world applications, including physics,[4,5] engineering,[6–8] biology,[9] finance[10] and many others. The complexity of nonlinear equations

$$f(\varepsilon) = 0 \tag{1}$$

makes them one of the most persistent problems in science and engineering.

Using a series of linear approximations, the Newton method[11,12] iteratively improves a first estimate for a solution to a nonlinear equation. It converges faster — often at a quadratic rate — to the exact solution through calculus-based iterations as follows:

$$\varepsilon_i^{[1]} = \varepsilon_i - \frac{f(\varepsilon_i)}{f'(\varepsilon_i)}. \tag{2}$$

Method (2) was developed in the late fourteenth century and has been widely used for many years to solve nonlinear equations. The fractional Newton technique, often referred to as the fractional variant of Newton's method, incorporates fractional calculus operations, such as fractional derivatives and integrals, into the iterative framework, thereby expanding the traditional Newton scheme. In scientific and engineering applications like viscoelastic materials, anomalous diffusion processes, and complex dynamical systems modeled by fractional differential equations offered by Pavani *et al.*[13] and Gasimov *et al.*,[14] this extension enables the effective handling of nonlinear equations with non-integer order derivatives. To deal with the memory effects and long-range interactions inherent in fractional models, fractional Newton-type algorithms provide improved convergence and stability compared to traditional numerical methods.[15] Recent studies have demonstrated the advantages of fractional approaches in fluid flow, biological domains and nonlinear wave propagation.[16,17] For example, studies on Casson nanofluid flow in stenotic arteries have demonstrated that fractional-order models may accurately reflect the rheological behavior of complicated fluids, as illustrated by Ramasekhar *et al.*[18] Furthermore, studies of soliton solutions of nonlinear optics have been conducted using fractional techniques, which have shed light on wave dynamics controlled by fractional differential equations.[19] The fractional Newton technique[20] has been used in numerical analysis to improve the precision and efficacy of resolving variational formulations with indefinite integrals and nonlinear eigenvalue problems.[21] Fractional-order modeling has also proven helpful for semi-analytical and numerical approaches to the production of entropy in hybrid nanofluid flows.[22] Utilization of non-singular fractional operators and Mittag-Leffler kernels in fuzzy variable-order differential equations has significantly increased solution accuracy.[23]

Mechanical oscillators, fractal-based structures used in mechanical systems, and elevator buffer designs are just a few of the numerous engineering applications that benefit tremendously from ongoing advances in fractional iterative techniques (see, for example, Refs. 24–26 and the references cited therein). To integrate theoretical advances with practical applications, researchers are continuing to examine new areas of nonlinear analysis using the fractional Newton technique. Through these activities, fractional calculus's growing importance in modern computing technologies is highlighted, paving the way for more exact and efficient numerical solutions in a variety of scientific and industrial sectors.

The fractional Newton approach shows promising convergence features and has been used to solve complex nonlinear equations where other methods fail to converge or produce exact solution. Torres-Hernandez *et al.*,[27] Akgül *et al.*,[28] Gajori *et al.*,[29] and Kumar *et al.*[30] describe a fractional version of the Newton technique with various fractional derivatives. Inspired by the previously mentioned techniques, the main objective of this study is to develop and analyze efficient fractional-order iterative techniques for resolving nonlinear equations, utilizing fractional calculus to improve precision and stability. The aim of the study is to find out how different fractional orders affect numerical stability, computing efficiency and convergence behavior.

The main contributions of this research work are as follows:

- Construction of a numerical scheme utilizing the principle of fractional calculus.
- Using a symbolic computational tool Mathematica 9 to determine the proposed scheme's convergence order.
- Perform a dynamic investigation of the proposed fractional technique based on fractal behavior.
- Computational analysis is thoroughly explored to determine this proposed fractional technique's convergence rate, efficiency and stability.

We now introduce the mathematical notations that will be used to describe the methods and the subsequent components.

Notations

ß	Fractional parameters
${}_{\check{c}}\eth^{ß}_{ß_1}$	Caputo fractional derivative
$\lceil$	Gamma function
$CK^{ß}$, $MK^{ß*}$	Existing methods
$MK^{ß}$	Newly developed scheme
ETe	Elapsed time
DPs	Divergence points
CPs	Convergence points
PCe	Percentage-convergence
ξ	Exact solution
e_i	Residual error
$\in$	Tolerance

Except for the Caputo derivative, all fractional-type derivatives e.g. Riemann–Liouville derivative, Grünwald–Letnikov derivative and Caputo–Fabrizio fractional derivative are failed to satisfy $[{}_{\check{c}}\eth^{ß}_{ß_1}](C) = 0$ if ß is not a natural number i.e.

$$[{}_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon) = \frac{1}{\lceil(n-ß)} \times \int_0^{\varepsilon} \left(\frac{\frac{d^n f(t)}{dt^n}}{(\varepsilon - t)^{ß-n+1}} dt \right), \tag{3}$$

where n is the smallest integer and $n \geq ß$. Therefore, if $f(\varepsilon) = C$, then $\frac{d^n f(\varepsilon)}{dt^n} = 0$, where C is constant. The Caputo derivative is commonly used for fractional-order modeling because it is consistent with classical derivatives and compatible with standard initial conditions. It also preserves stability and accuracy while remaining consistent with the formulation of real-world problems. Other fractional derivatives, while mathematically accurate, frequently lack the practical flexibility and alignment with physical models. Therefore, we will cover some basic ideas in fractional calculus as well as the fractional iterative approach for solving nonlinear equations using Caputo-type fractional derivatives.

Definition. The function defined below is known as the Gamma function[31] or generalized factorial function, as

$$\lceil(\varepsilon) = \int_0^{+\infty} u^{\varepsilon-1} e^{-u} du, \tag{4}$$

where $\varepsilon > 0$, $\lceil(1) = 1$ and $\lceil(n+1) = n!$ where $n \in N$.

Definition. Caputo fractional derivative $[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon)^{32}$ of order ß is defined as

$$[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon) = \begin{cases} \left(\dfrac{1}{\lceil(m-\ss)}\displaystyle\int_0^{\varepsilon}\dfrac{d^m f(t)}{dt^m}\dfrac{1}{(\varepsilon-t)^{\ss-m+1}}dt, \ss\notin\mathbb{N}\right), \\ \dfrac{d^{m-1}f(\varepsilon)}{dt^{m-1}}, \quad \ss = m-1\in\mathbb{N}\cup\{0\}, \end{cases} \tag{5}$$

where $\lceil(\varepsilon)$ is a gamma function with $\varepsilon > 0$.

Theorem 1. *Suppose* $[{}_{\check{c}}\eth^{\gamma\beta}_{\beta_1}]f(\varepsilon) \in \check{c}([\beta_1,\beta_2])$ *for* $\gamma = 1,\ldots,n+1$ *where* $\ss\in(0,1]$, *then generalized Taylor formula*[33,34] *is*

$$f(\varepsilon) = \left(\begin{array}{c} \displaystyle\sum_{i=0}^{n}[{}_{\check{c}}\eth^{i\beta}_{\beta_1}]f(\beta_1)\frac{(\varepsilon-\beta_1)^{i\beta}}{\lceil(i\beta+1)} \\ +[{}_{\check{c}}\eth^{(n+1)\beta}_{\beta_1}]f(\xi)\dfrac{(\varepsilon-\beta_1)^{(n+1)\beta}}{\lceil((n+1)\beta+1)} \end{array}\right) \tag{6}$$

and

$$\beta_1 \le \xi \le \varepsilon, \quad \forall\varepsilon\in(\beta_1,\beta_2] \tag{7}$$

and

$$[{}_{\check{c}}\eth^{n\beta}_{\beta_1}] = \left(\begin{array}{c} [{}_{\check{c}}\eth^{\beta}_{\beta_1}] \\ [{}_{\check{c}}\eth^{\beta}_{\beta_1}]\cdots[{}_{\check{c}}\eth^{\beta}_{\beta_1}](n\text{-times}) \end{array}\right). \tag{8}$$

Consider the Caputo-type Taylor evolution of $f(\varepsilon)$ near $\ss_1 = \xi$ as

$$f(\varepsilon) = \left(\begin{array}{c} \dfrac{[{}_{\check{c}}\eth^{1\ss}_{\xi}]f(\xi)}{\lceil(+1)}(\varepsilon-\xi)^{\ss} \\ +\dfrac{[{}_{\check{c}}\eth^{2\ss}_{\xi}]f(\xi)}{\lceil(2\ss+1)}(\varepsilon-\xi)^{2\ss} \\ +O(\varepsilon-\xi)^{3\ss} \end{array}\right). \tag{9}$$

Taking $\frac{[{}_{\check{c}}\eth^{1\ss}_{\xi}]f(\xi)}{\lceil(\ss+1)}$ common, we have

$$f(\varepsilon) = \left(\begin{array}{c} \dfrac{[{}_{\check{c}}\eth^{1\ss}_{\xi}]f(\xi)}{\lceil(\ss+1)}\left[\begin{array}{c}(\varepsilon-\xi)^{\ss} \\ +\check{c}_2(\varepsilon-\xi)^{2\ss}\end{array}\right] \\ +O(\varepsilon-\xi)^{3\ss} \end{array}\right), \tag{10}$$

where $\check{c}_\gamma = \frac{\lceil(\ss+1)}{\lceil(\gamma\ss+1)}\frac{[{}_{\check{c}}\eth^{\gamma\ss}_{\xi}]f(\xi)}{[{}_{\check{c}}\eth^{\ss}_{\xi}]f(\xi)}, \gamma \ge 2$. Corresponding Caputo-type derivative of $f(\varepsilon)$ around ξ is

$$[{}_{\check{c}}\eth^{\ss}_{\xi}]f(\varepsilon) = \left(\dfrac{[{}_{\check{c}}\eth^{1\ss}_{\xi}]f(\xi)}{\lceil(\ss+1)}\left[\begin{array}{c}\lceil(\ss+1) \\ +\dfrac{\lceil(2\ss+1)}{\lceil(\ss+1)}\check{c}_2(\varepsilon-\xi)^{\ss}\end{array}\right]\right) + O(\varepsilon-\xi)^{2\ss}. \tag{11}$$

They are used in the convergence analysis of the proposed method.

Using the Caputo-type fractional version of classical Newton's method, Candelario *et al.*[35] presented the following variant:

$$\varepsilon_i^{[1]} = \varepsilon_i - \left(\lceil(\ss+1)\frac{f(\varepsilon_i)}{[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon_i)}\right)^{1/\ss}, \tag{12}$$

where $[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon_i) \approx [{}_{\check{c}}\eth^{\ss}_{\xi}]f(\xi)$ for any $\ss\in\mathbb{R}$. The following error equation is satisfied by the fractional Newton method's order of convergence, which is $\ss+1$ (abbreviated as CK $^{\ss*}$).

$$e_i^{[1]} = \frac{\lceil(2\ss+1)-\lceil^2(\ss+1)}{\ss\lceil^2(\ss+1)}\check{c}_2 e_i^{\ss+1} + O(e_i^{2\ss+1}), \tag{13}$$

where $e_i^{[1]} = \varepsilon_i^{[1]} - \xi$ and $e_i = \varepsilon_i - \xi$ and $\check{c}_\gamma = \frac{\lceil(\ss+1)}{\lceil(\gamma\ss+1)}\frac{[{}_{\check{c}}\eth^{\gamma\ss}_{\xi}]f(\xi)}{[{}_{\check{c}}\eth^{\ss}_{\xi}]f(\xi)}, \gamma \ge 2$ and and $(\lceil(.))^n = \lceil^n(.)$.

Shams *et al.*[36] proposed the following single-step fractional iterative method as

$$\varepsilon_i^{[1]} = \varepsilon_i - \left(\begin{array}{c} \lceil(\ss+1)\dfrac{f(\varepsilon_i)}{[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon_i)} \\ \left[\dfrac{1}{1-\alpha\frac{f(\varepsilon_i)}{1+f(\varepsilon_i)}}\right] \end{array}\right)^{1/\ss}. \tag{14}$$

The order of convergence of (14) technique is $\ss+1$ (abbreviated as MK$^{\ss*}$), which satisfies the following error equation:

$$e_i^{[1]} = \left(\frac{\left(\begin{array}{c}(\alpha+\check{c}_2)\lceil^2(\ss+1) \\ -\check{c}_2\lceil(2\ss+1)\end{array}\right)}{\ss\lceil(\ss+1)}\check{c}_2 e_i^{\ss+1} + O(e_i^{2\ss+1})\right), \tag{15}$$

where $e_i^{[1]} = \varepsilon_i^{[1]} - \xi$ and $e_i = \varepsilon_i - \xi$. The following Caputo-type fractional version of (2) was proposed in Ref. 35 as

$$\varepsilon_i^{[2]} = \varepsilon_i^{[1]} - \left(\begin{array}{c} \lceil(\ss+1) \\ \dfrac{f(\varepsilon_i^{[1]})}{[{}_{\check{c}}\eth^{\ss}_{\ss_1}]f(\varepsilon_i)} \end{array}\right)^{1/\ss}, \tag{16}$$

where $\varepsilon_i^{[1]} = \varepsilon_i - (\lceil(ß+1)\frac{f(\varepsilon_i)}{[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i)})^{1/ß}$, $[_{\check{c}}\eth^{ß}_{ß_1}] \times f(\varepsilon_i) \approx [_{\check{c}}\eth^{ß}_{\xi}]f(\xi)$ for any $ß \in \mathbb{R}$. The order of convergence of the (16) technique is 2ß+1 (abbreviated as $CK^{ß}$), which satisfies the following error equation:

$$e_i^{[1]} = \begin{pmatrix} -\frac{2\lceil(ß+1) - \lceil(ß+1)}{ß^2\lceil^2(ß+1)} B^* \check{c}_2^2 e_i^{2ß+1} \\ + O(e_i^{ß^2+2ß+1}) \end{pmatrix}, \tag{17}$$

where $B^* = \frac{\lceil^2(ß+1) - \lceil(2ß+1)}{\lceil^2(ß+1)}$, $e_i^{[1]} = \varepsilon_i^{[1]} - \xi$ and $e_i = \varepsilon_i - \xi$.

2. FRACTIONAL SCHEME CONSTRUCTION AND ANALYSIS

Consider the Ostrowski's method[37] as

$$\varepsilon_i^{[2]} = \varepsilon_i^{[1]} - \left(\frac{f(\varepsilon_i^{[1]})}{f'(\varepsilon_i)}\left(\frac{1}{1-2(\frac{f(\varepsilon_i^{[1]})}{f(\varepsilon_i)})}\right)\right), \tag{18}$$

where $\varepsilon_i^{[1]} = \varepsilon_i - \left(\frac{f(\varepsilon_i)}{f'(\varepsilon_i)}\right)$. The fractional version of (18) is given as

$$\varepsilon_i^{[2]} = \varepsilon_i^{[1]} - \begin{pmatrix} \lceil(ß+1)\frac{f(\varepsilon_i^{[1]})}{[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i)} \\ \left(\frac{1}{1-2(\frac{f(\varepsilon_i^{[1]})}{f(\varepsilon_i)})}\right) \end{pmatrix}^{1/ß}, \tag{19}$$

where

$$\varepsilon_i^{[1]} = \varepsilon_i - \begin{pmatrix} \lceil(ß+1)\frac{f(\varepsilon_i)}{[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i)} \\ \left(\frac{1}{1-\alpha(\frac{f(\varepsilon_i)}{1+(f(\varepsilon_i))})}\right) \end{pmatrix}^{1/ß}.$$

We abbreviated this method by $MK^{ß}$.

Convergence Analysis

For iterative schemes (19), we prove the following theorem to establish its order of convergence.

Theorem. *Let*

$$f : \eth \subseteq \mathbb{R} \to \mathbb{R}$$

be the continuous function with $[_{\check{c}}\eth^{\gamma ß}_{ß_1}]f(\varepsilon)$ *of order* $\gamma ß$ *for any* $\gamma \geq 0$ *and* $ß \in (0,1]$ *containing exact root* ξ *of* $f(\varepsilon)$ *Furthermore, for a sufficiently near-total starting value* ε_0, *the convergence order of the Caputo-type fractional iterative schemes*

$$\varepsilon_i^{[2]} = \varepsilon_i^{[1]} - \begin{pmatrix} \lceil(ß+1)\frac{f(\varepsilon_i^{[1]})}{[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i)} \\ \left(\frac{1}{1-2(\frac{f(\varepsilon_i^{[1]})}{f(\varepsilon_i)})}\right) \end{pmatrix}^{1/ß}$$

is least $2ß+1$ *and the error equation is as*

$$e_i^{[1]} = \begin{pmatrix} \begin{pmatrix} \frac{3(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2^2}{ß\lceil(ß)\sqrt{\pi}} \\ -\frac{(2^{ß})^4\lceil^2(ß+\frac{1}{2})\,\check{c}_2^2}{ß^2(\lceil(ß))^2\pi} - 2\check{c}_2^2 \end{pmatrix} e_i^{2ß+1} \\ + O(e_i^{3ß+1}) \end{pmatrix},$$

where $\check{c}_\gamma = \frac{\lceil(ß+1)}{\lceil(\gamma ß+1)}\frac{[_{\check{c}}\eth^{\gamma ß}_{\xi}]f(\xi)}{[_{\check{c}}\eth^{ß}_{\xi}]f(\xi)}, \gamma \geq 2.$

Proof. Let ξ be a root of f and $\varepsilon_i = \xi + e_i$. By Taylor's series expansion of $f(\varepsilon_i)$ and $[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i)$ around $\varepsilon = \xi$, taking $f(\xi) = 0$, we get

$$f(\varepsilon_i) = \begin{bmatrix} \frac{[_{\check{c}}\eth^{1ß}_{\xi}]f(\xi)}{\lceil(ß+1)} \\ [e_i^{ß} + \check{c}_2 e_i^{2ß} + \check{c}_3 e_i^{3ß}] \end{bmatrix} + O(e_i^{4ß})$$

and

$$[_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i) = \begin{pmatrix} \frac{[_{\check{c}}\eth^{1ß}_{\xi}]f(\xi)}{\lceil(ß+1)} \begin{bmatrix} \lceil(ß+1) + \frac{\lceil(2ß+1)}{\lceil(ß+1)}\check{c}_2 e_i^{ß} \\ + \frac{\lceil(3ß+1)}{\lceil(2ß+1)}\check{c}_3 e_i^{2ß} \end{bmatrix} \\ + O(e_i^{3ß}) \end{pmatrix},$$

$$\begin{aligned} ([_{\check{c}}\eth^{ß}_{ß_1}]f(\varepsilon_i))^{-1} &= \frac{1}{\lceil(ß+1)} - \frac{\lceil(2ß+1)}{\lceil(ß+1)}\check{c}_2 e_i^{ß} \\ &+ \frac{(-\frac{\lceil(3ß+1)}{\lceil(ß+1)\lceil(2ß+1)} + \frac{(\lceil(2ß+1))^2}{(\lceil(ß+1))^4})}{\lceil(ß+1)}\check{c}_3 e_i^{2ß} \\ &+ \begin{pmatrix} \frac{\lceil(3ß+1)}{\lceil(ß+1)}\check{c}_2\check{c}_3 \\ \frac{-\frac{(\lceil(2ß+1))^3\,\check{c}_2^2 - \lceil(3ß+1) \times (\lceil(ß+1))^3\,\check{c}_2\check{c}_3}{(\lceil(ß+1))^6}}{\lceil(ß+1)} \end{pmatrix} e_i^{3ß} \\ &+ O(e_i^{4ß}). \end{aligned}$$

Dividing (16) by (17), we have

$$\frac{f(\varepsilon_i)}{[{}_{\check{c}}Ↄ^{ß}_{ß_1}]f(\varepsilon_i)} = \frac{1}{ß\lceil(ß)}e_i^{ß} + \left(\frac{(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2}{ß^2(\lceil(ß))^2\sqrt{\pi}} + \frac{\check{c}_2}{ß\lceil(ß)}\right)e_i^{2ß} + \left(\begin{array}{l}\frac{(2^{ß})^4(\lceil(ß+\frac{1}{2}))^2\check{c}_2^2}{ß^3(\lceil(ß))^3\pi} \\ -\frac{(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2^3}{ß^2(\lceil(ß))^2\sqrt{\pi}} \\ -\frac{1}{2}\frac{\check{c}_3\left(3^{ß}\sqrt{3}\lceil(ß+\frac{1}{3})\;\lceil(ß+\frac{2}{3})\right)}{ß^2(\lceil(ß))^2\times\sqrt{\pi}(2^{ß})^2\lceil(ß+\frac{1}{2})} \\ +\frac{\check{c}_3}{ß\lceil(ß)}\end{array}\right)e_i^{3ß} + O(e_i^{3ß}),$$

where $\lceil(\frac{1}{ß}+1) = \frac{1}{ß}\lceil(\frac{1}{ß})$.

$$\begin{aligned}\varepsilon_i^{[1]} - \xi &= \varepsilon_i - \xi - \left(\lceil(ß+1)\frac{f(\varepsilon_i)}{[{}_{\check{c}}Ↄ^{ß}_{ß_1}]f(\varepsilon_i)}\left(\frac{1}{1-\alpha\left(\frac{f(\varepsilon_i)}{1+(f(\varepsilon_i))}\right)}\right)\right)^{1/ß} \\ &= \left(\frac{(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2 - \check{c}_2ß^2(\lceil(ß))^2\sqrt{\pi}}{ß^2(\lceil(ß))^2\sqrt{\pi}}\right)e_i^{ß+1} \\ &\quad + \left[\begin{array}{l}\frac{(2^{ß})^4\left(\lceil(ß+\frac{1}{2})\right)^2\check{c}_2^2}{ß^3\left(\lceil(ß)\right)^3\pi} \\ +\frac{(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2^3}{ß^2\left(\lceil(ß)\right)^2\sqrt{\pi}} \\ +\frac{1}{2}\frac{\check{c}_3\left(3^{ß}\sqrt{3}\lceil(ß+\frac{1}{3})\times\lceil(ß+\frac{2}{3})\right)}{\left(ß^2\left(\lceil(ß)\right)^2\sqrt{\pi}(2^{ß})^2\times\lceil(ß+\frac{1}{2})\right)} - \frac{\check{c}_3}{ß\lceil(ß)}\end{array}\right]e_i^{2ß+1} \\ &\quad + O(e_i^{3ß+1}).\end{aligned}$$

Using generalized binomial theorem $(\varepsilon+y)^t = \sum_{i=0}^{+\infty}\binom{t}{i}\varepsilon^{t-i}y^i$ where $\binom{t}{i} = \frac{\lceil(t+1)}{i!\lceil(t-i+1)}$. Expanding $f(\varepsilon_i^{[1]})$ around ξ, we have

$$f\left(\varepsilon_i^{[1]}\right) = \check{c}_2\left(\frac{\left(ß\lceil(ß)\sqrt{\pi} - (2^{ß})^2\lceil(ß+\frac{1}{2})\right)}{ß\lceil(ß)\sqrt{\pi}}\right)e_i^{ß+1} - \frac{1}{2}A_{11} * B_{11}e_i^{2ß+1} + \cdots,$$

where

$$A_{11} = \frac{1}{\left(ß^2\left(\lceil(ß)\right)^2\pi^{\frac{3}{2}}\left(2^{ß}\right)^2\times\lceil\left(ß+\frac{1}{2}\right)\right)},$$

$$B_{11} = \left[\begin{array}{l}\left(2(2^{ß})^6\left(\lceil\left(ß+\frac{1}{2}\right)\right)^3\times\check{c}_2^2\sqrt{\pi} - 2(2^{ß})^4\times\left(\lceil\left(ß+\frac{1}{2}\right)\right)^2\check{c}_2^2ß\lceil(ß)\pi\right) \\ +\left(2\check{c}_3(ß\lceil(ß))^2\times\pi^{\frac{3}{2}}(2^{ß})^2\lceil\left(ß+\frac{1}{2}\right)\right) \\ -\left(\check{c}_3(3^{ß})^3\sqrt{3}\lceil\left(ß+\frac{1}{3}\right)\times\lceil\left(ß+\frac{2}{3}\right)ß\lceil(ß)\pi\right)\end{array}\right].$$

Then

$$\frac{f(\varepsilon_i^{[1]})}{f(\varepsilon_i)} = \left(\frac{\left(2^{ß}\lceil(ß+\frac{1}{2})\right)^2\check{c}_2}{ß^2\left(\lceil(ß)\right)^2\sqrt{\pi}} - \frac{\check{c}_2}{ß\lceil(ß)}\right)e_i^{ß+1} + \left(\begin{array}{l}\frac{(2^{ß})^4(\lceil(ß+\frac{1}{2}))^2\check{c}_2^2}{ß^3(\lceil(ß))^3\pi} \\ +\left(\frac{\left((2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2^3\right)}{ß^2\left(\lceil(ß)\right)^2\sqrt{\pi}}\right) \\ +\frac{1}{2}\frac{\left(\check{c}_3\left(3^{ß}\sqrt{3}\lceil(ß+\frac{1}{3})\times\lceil(ß+\frac{2}{3})\right)\right)}{\left(ß^2\left(\lceil(ß)\right)^2\sqrt{\pi}\left(2^{ß}\right)^2\times\lceil(ß+\frac{1}{2})\right)} \\ -\frac{\check{c}_3}{ß\lceil(ß)}\end{array}\right)e_i^{2ß+1} + O(e_i^{3ß+1}),$$

$$f(\varepsilon_i) - 2f(\varepsilon_i^{[1]}) = e_i^{ß} + \left(3\check{c}_2 - \frac{2(2^{ß})^2\lceil(ß+\frac{1}{2})\check{c}_2}{ß\lceil(ß)\sqrt{\pi}}\right)e_i^{ß+1}$$

$$+\left(\left(\begin{array}{l}3\check{c}_3+\dfrac{2(2^{\text{ß}})^4(\lceil(\text{ß}+\frac{1}{2}))^2\check{c}_2^2}{\text{ß}^2\left(\lceil(\text{ß})\right)^2\pi}\\ -\dfrac{2\left(2^{\text{ß}}\right)^2\lceil(\text{ß}+\frac{1}{2})\check{c}_2^2}{\text{ß}\left(\lceil(\text{ß})\right)\sqrt{\pi}}\end{array}\right)-\dfrac{\check{c}_3\left(\begin{array}{c}(3^{\text{ß}})^3\sqrt{3}\lceil(\text{ß}+\frac{1}{3})\\ \times\lceil(\text{ß}+\frac{2}{3})\end{array}\right)}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2})}\right)e_i^{2\text{ß}+1}$$

$$+O(e_i^{3\text{ß}+1}),$$

$$\frac{f(\varepsilon_i)}{f(\varepsilon_i)-2f(\varepsilon_i^{[1]})}=1+\left(\frac{\left(2(2^{\text{ß}})^2\lceil\left(\text{ß}+\frac{1}{2}\right)\right)^2\check{c}_2}{\text{ß}\left(\lceil(\text{ß})\right)\sqrt{\pi}}-2\check{c}_2\right)e_i^{\text{ß}}$$

$$+\left(\begin{array}{l}-\dfrac{8((2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))\check{c}_2^2}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}}\\ +\dfrac{(2(2^{\text{ß}})^4\lceil(\text{ß}+\frac{1}{2}))^2\check{c}_2^2}{\text{ß}^2(\lceil(\text{ß}))^2\pi}+\\ +\dfrac{\check{c}_3((3^{\text{ß}})^3\sqrt{3}\lceil(\text{ß}+\frac{1}{3})\lceil(\text{ß}+\frac{2}{3}))}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2})}\\ +6\check{c}_2^2-2\check{c}_3\end{array}\right)e_i^{\text{ß}+1}$$

$$+O(e_i^{2\text{ß}+1}),$$

$$\frac{f(\varepsilon_i)}{f(\varepsilon_i)-2f(\varepsilon_i^{[1]})}\frac{f(\varepsilon_i^{[1]})}{[\check{c}\text{Ɔ}^{\text{ß}}_{\text{ß}_1}]f(\varepsilon_i)}=\left(\frac{((2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))^2\check{c}_2}{\text{ß}^2(\lceil(\text{ß}))^2\sqrt{\pi}}-\frac{2\check{c}_2}{\text{ß}(\lceil(\text{ß}))}\right)e_i^{\text{ß}+1}$$

$$+\left(\begin{array}{l}\dfrac{\check{c}_3}{\text{ß}\left(\lceil(\text{ß})\right)}+\dfrac{2\check{c}_2^2}{\text{ß}\left(\lceil(\text{ß})\right)}\\ -\dfrac{(2(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))\check{c}_2}{\text{ß}^2(\lceil(\text{ß}))^2\sqrt{\pi}}\\ +\dfrac{1}{2}\dfrac{\left(\check{c}_3\left(\begin{array}{c}(3^{\text{ß}})^3\sqrt{3}\lceil(\text{ß}+\frac{1}{3})\\ \times\lceil(\text{ß}+\frac{2}{3})\end{array}\right)\right)}{\left(\begin{array}{c}\text{ß}\left(\lceil(\text{ß})\right)\sqrt{\pi}\left(2^{\text{ß}}\right)^2\\ \times\lceil(\text{ß}+\frac{1}{2})\end{array}\right)}\end{array}\right)e_i^{2\text{ß}+1}$$

$$+O(e_i^{3\text{ß}+1}),$$

$$\lceil(\text{ß}+1)\frac{f(\varepsilon_i)}{f(\varepsilon_i)-2f(\varepsilon_i^{[1]})}\frac{f(\varepsilon_i^{[1]})}{[\check{c}\text{Ɔ}^{\text{ß}}_{\text{ß}_1}]f(\varepsilon_i)}=\left(\frac{((2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))^2\check{c}_2}{\text{ß}\left(\lceil(\text{ß})\right)\sqrt{\pi}}-\check{c}_2\right)e_i^{\text{ß}+1}$$

$$+\left(\begin{array}{l}-\dfrac{\check{c}_3}{1}+\dfrac{2\check{c}_2^2}{1}\\ -\dfrac{(2(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))\check{c}_2^2}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}}\\ +\dfrac{1}{2}\dfrac{\check{c}_3((3^{\text{ß}})^3\sqrt{3}\lceil(\text{ß}+\frac{1}{3})\lceil(\text{ß}+\frac{2}{3}))}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2})}\end{array}\right)e_i^{2\text{ß}+1}$$

$$+\left(\begin{array}{l}\dfrac{\check{c}_2^3(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2})^3\check{c}_2\check{c}_3}{\text{ß}^3(\lceil(\text{ß}))^3\pi^{\frac{3}{2}}}\\ -\dfrac{4(2^{\text{ß}})^4\lceil(\text{ß}+\frac{1}{2})^2\check{c}_2^3}{\text{ß}^{02}(\lceil(\text{ß}))^2\pi}\\ -\dfrac{\check{c}_3\check{c}_2\left(\begin{array}{c}(3^{\text{ß}})^3\sqrt{3}\lceil(\text{ß}+\frac{1}{3})\\ \times\lceil(\text{ß}+\frac{2}{3})\end{array}\right)}{\text{ß}\left(\lceil(\text{ß})\right)\sqrt{\pi}\left(2^{\text{ß}}\right)^2\lceil(\text{ß}+\frac{1}{2})}\\ -\check{c}_4-\dfrac{(2(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2}))\check{c}_3\check{c}_2}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}}\\ +4\check{c}_2\check{c}_3-5\check{c}_2^3\\ +\dfrac{8(2^{\text{ß}})^2\lceil(\text{ß}+\frac{1}{2})_3\check{c}_2^3}{\text{ß}(\lceil(\text{ß}))\sqrt{\pi}}\end{array}\right)(e_i^{3\text{ß}+1})$$

$$+O(e_i^{4\text{ß}+1}),$$

$$\varepsilon_i^{[2]}-\xi=\varepsilon_i^{[1]}-\xi-\left(\begin{array}{c}\left(\begin{array}{c}\lceil(\text{ß}+1)\\ \dfrac{f(\varepsilon_i^{[1]})}{[\check{c}\text{Ɔ}^{\text{ß}}_{\text{ß}_1}]f(\varepsilon_i)}\end{array}\right)\\ \left(\dfrac{1}{1-2\left(\frac{f(\varepsilon_i^{[1]})}{f(\varepsilon_i)}\right)}\right)\end{array}\right)^{1/\text{ß}},$$

$$e_i^{[1]}=\left(\begin{array}{l}\dfrac{\left(3(2^{\text{ß}})^2\lceil\left(\text{ß}+\frac{1}{2}\right)\check{c}_2^2\right)}{\text{ß}\lceil(\text{ß})\sqrt{\pi}}\\ -\dfrac{\left((2^{\text{ß}})^4\lceil^2\left(\text{ß}+\frac{1}{2}\right)\check{c}_2^2\right)}{(\text{ß}^2(\lceil(\text{ß}))^2\pi)}-2\check{c}_2^2\end{array}\right)e_i^{2\text{ß}+1}$$

$$+\ O(e_i^{3\text{ß}+1}).$$

Hence we prove the theorem. □

3. FRACTAL ANALYSIS

The fractal behavior of iterative approaches for solving nonlinear equations is an astonishing phenomenon that reveals the complexities of numerical computations. Fractal-based approaches enhance iterative schemes' accuracy, stability and efficiency in complex nonlinear engineering problems, allowing the scheme to choose better initial starting values. In order to better reflect the complex features of nonlinear systems, adaptive fractional numerical techniques dynamically modify fractional orders, incorporating fractal analysis. This flexibility improves convergence by refining initial approximations, resulting in greater accuracy and stability. These strategies employ fractal features to improve guess assumptions and directional changes, reducing sensitivity to poor starting guesses and ensuring faster convergence to the exact solution. Iterative algorithms utilize repeated steps to approximate the roots of nonlinear equations, frequently displaying fractal-like self-similar patterns. Through iterative refinement of estimates, these approaches go through zones of convergence and divergence, revealing complex structures that display self-similarity on many scales. The Mandelbrot set, a classic example of fractals in mathematics, graphically depicts this concept, highlighting the complex boundaries between convergence and divergence for a specific iterative function. For details on the dynamical behavior of the iterative methods, one can consult Refs. 38–40. Using iterative methods to solve nonlinear equations in fractal environments, the stability and accuracy of numerical solutions are affected by varying fractional orders. Lower fractional orders improve stability by including historical dependency and memory effects, making the solution more resistant to disturbances. Higher fractional orders, on the other hand, boost accuracy by more accurately modeling complex dynamics, but if not handled appropriately, they may result in higher processing costs and even instability. To ensure appropriate numerical approximations in nonlinear processes, stability and precision must be balanced when utilizing fractional order, as indicated in Tables 1–3. In nonlinear engineering applications, fractal geometry improves modeling of irregular domains by more accurately capturing scale-invariant structures, complexity and self-similarity as shown in Figs. 1–3, respectively.

Grid 600×600, with its center at the origin, is utilized to generate basins of attraction, and $360,000$ points in total are used to create the dynamical planes within the square $[-2,2]\times[-2,2]^2 \in \mathbb{C}$. We designate a color to each root of $f(\varepsilon)=0$, where the associated orbit of the iterative methods begins and, for varying values of ß, converges to a fixed point. Color map taken as Jet. As a stopping criterion,

Table 1 **Fractal Analysis of Numerical Schemes CK^{ß}, MK^{ß} and MK^{ß*} for (20).**

	CK^{ß}		MK^{ß}		MK^{ß*}
Figs.	**1a**	**1b**	**1c**	**1d**	**1e**
ß	0.80	0.99	0.80	0.99	1.00
ETe	23.234654	5.234543	2.34123	16.2342	2.123
DPs	5423	4564	3678	2543	1546
CPs	354,577	355,436	356,322	357,457	3584
PCe	98.45	98.7	98.97	99.29	99.57

Table 2 **Fractal Analysis of Numerical Schemes CK^{ß}, MK^{ß} and MK^{ß*} for (22).**

	CK^{ß}		MK^{ß}		MK^{ß*}
Figs.	**1a**	**1b**	**1c**	**1d**	1e
ß	0.80	0.99	0.80	0.99	1.00
ETe	33.204	15.2343	20.343	10.0042	9.102
DPs	8923	5964	4678	3543	2546
CPs	351,077	354,036	355,322	356,457	357,454
PCe	97.52	98.34	98.77	98.72	99.29

Table 3 **Fractal Analysis of Numerical Schemes CK^{ß}, MK^{ß} and MK^{ß*} for (24).**

	CK^{ß}		MK^{ß}		MK^{ß*}
Figs.	**1a**	**1b**	**1c**	**1d**	1e
ß	0.80	0.99	0.80	0.99	1.00
ETe	53.234654	35.234543	32.34123	20.2342	25.123
DPs	6723	3564	2678	1543	2446
CPs	353,277	356,436	357,322	358,457	35,755
PCe	98.13	97.01	99.22	99.57	98.3

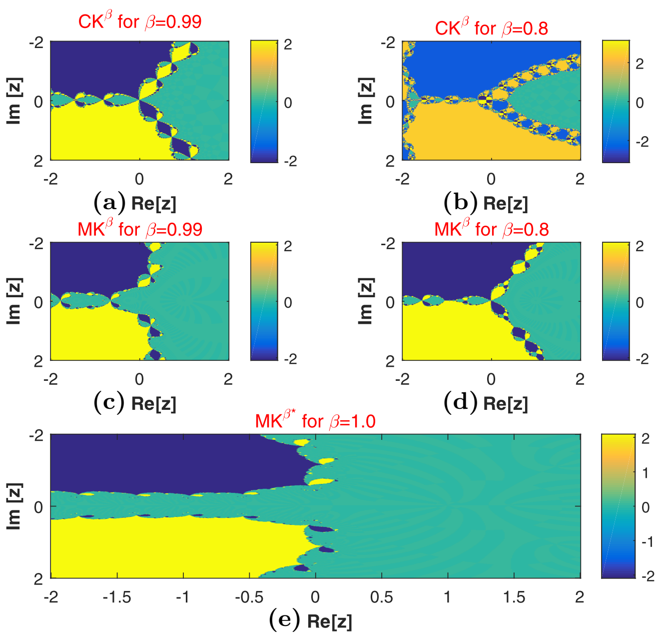

Fig. 1 Fractal behavior of CK^{ß}, MK^{ß} and MK^{ß*} for solving (20) for different values of ^{ß*}.

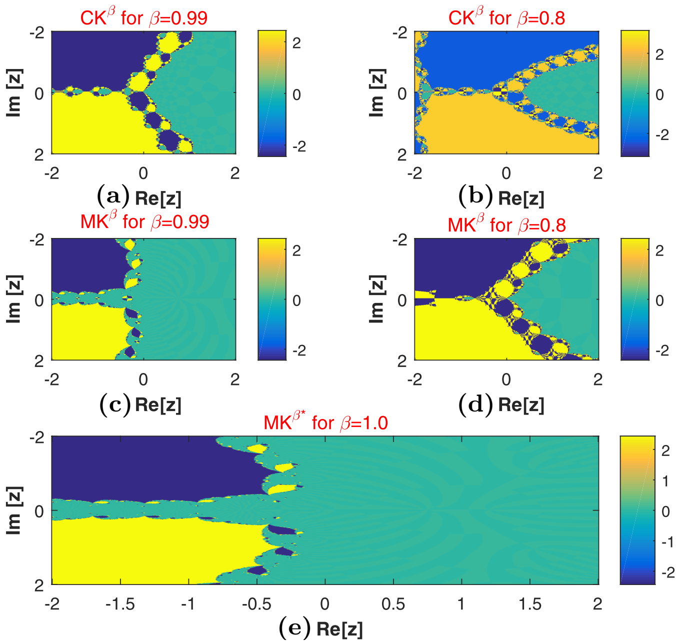

Fig. 2 Fractal behavior of CK^{β}, MK^{β} and $\text{MK}^{\beta*}$ for solving (22) for different values of $\beta*$.

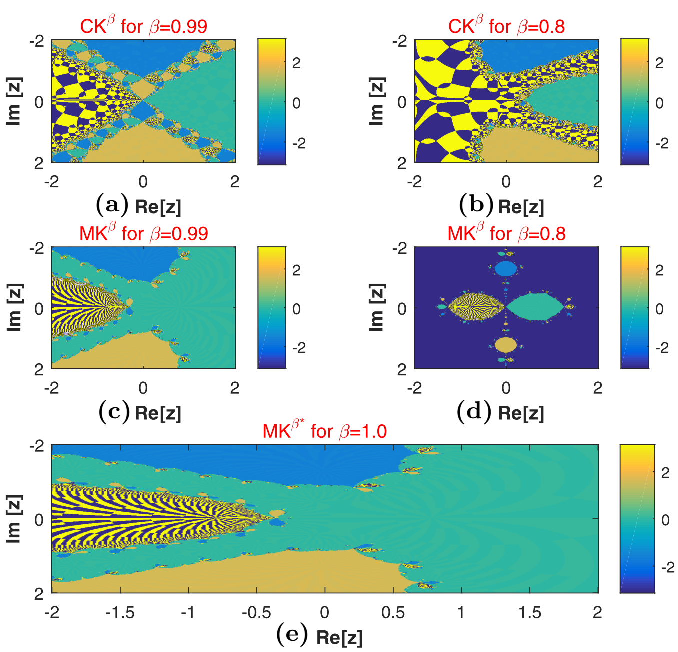

Fig. 3 Fractal behavior of CK^{β}, MK^{β} and $\text{MK}^{\beta*}$ for solving (24) for different values of $\beta*$.

we select $|f(\varepsilon_i)| < 0.0001$, and 50 is the maximum number of iterations. Consider the following function:

$$f(\varepsilon) = \varepsilon^3 - 1, \tag{20}$$

with exact roots are $\xi_1 = 1, \xi_{2,3} = \frac{1\pm\sqrt{3}i}{2}$. The corresponding Caputo fractional derivative is given as

$$\left[{}_{\tilde{c}}\mathfrak{D}^{\beta}_{\beta_1}\right] f(\varepsilon) = \frac{\Gamma(4)}{\Gamma(4-\beta)}\varepsilon^{3-\beta} - \frac{1}{\Gamma(1-\beta)}\varepsilon^{-\beta}. \tag{21}$$

Fractal analysis generated by iterative scheme for (20) is given in Table 1.

Consider another nonlinear function

$$f(\varepsilon) = \varepsilon^3 + \varepsilon^2 - 1, \tag{22}$$

with exact roots are $0.7, -0.8 + 0.7i, -0.8 - 0.7i$. The corresponding Caputo fractional derivative is given as

$$\left[{}_{\tilde{c}}\mathfrak{D}^{\beta}_{\beta_1}\right] f(\varepsilon) = \frac{\Gamma(4)}{\Gamma(4-\beta)}\varepsilon^{3-\beta} + \frac{\Gamma(3)}{\Gamma(3-\beta)}\varepsilon^{2-\beta} - \frac{1}{\Gamma(1-\beta)}\varepsilon^{-\beta}. \tag{23}$$

Fractal analysis generated by iterative scheme for (22) is given in Table 2.

Consider another nonlinear function

$$f(\varepsilon) = \varepsilon^4 - 1, \tag{24}$$

with exact roots are $\xi_1 = \pm 1, \xi_{2,3} = \pm i$. The corresponding Caputo fractional derivative is given as

$$\left[{}_{\tilde{c}}\mathfrak{D}^{\beta}_{\beta_1}\right] f(\varepsilon) = \frac{\Gamma(5)}{\Gamma(5-\beta)}\varepsilon^{4-\beta} - \frac{1}{\Gamma(1-\beta)}\varepsilon^{-\beta}. \tag{25}$$

Fractal analysis generated by iterative scheme for (24) is given in Table 3.

Tables 1–3 clearly show that in terms of elapsed time (ETe), divergence points (DPs), convergence points (CPs) and percentage-convergence (PCe), the family of fractional numerical scheme MK^{β} is better than CK^{β} and $\text{MK}^{\beta*}$.

4. NUMERICAL RESULTS

The following terminating criteria of the computer algorithm used in Maple 18 are used to examine several engineering applications in this part in order to demonstrate their effectiveness and stability:

$$\text{(i)} \quad e_i = |\varepsilon_i^{[1]} - \varepsilon_i| < \in, \quad \text{(ii)} \quad e_i = |f(\varepsilon_i^{[1]})| < \in, \tag{26}$$

where e_i represents the error in iterative scheme and $\in = 10^{-15}$.

Example 4.1. Civil Engineering Application
Figure shows an employee of "Down to the Toilet Company" who creates floats for $\text{A}^{[1]}\text{A}^{[2]}\text{A}^{[3]41}$ commodes. The floating ball has a specific gravity of 0.6 N and a radius of 5.5 cm. The depth ε in meters at

which the ball is submerged in water can be calculated as

$$\varepsilon^3 - 0.165\varepsilon^2 + 3.993 \times 10^{-4} = 0 \quad (27)$$

or

$$f(\varepsilon) = \varepsilon^3 - 0.165\varepsilon^2 + 3.993 \times 10^{-4}. \quad (28)$$

Nonlinear equation (28) has three real roots $\xi_1 = -0.04374, \xi_2 = 0.1464, \xi_3 = 0.0624$ as shown in the figure. Our desired root is $\xi_3 = 0.0624$. The Caputo-type derivative of (28) are

$$\left[{}_{\breve{c}}\eth^{ß}_{ß_1}\right] f(\varepsilon) = \begin{pmatrix} \frac{\lceil(4)}{\lceil(4-ß)}\varepsilon^{3-ß} \\ -0.165\frac{\lceil(3)}{\lceil(3-ß)}\varepsilon^{2-ß} \\ +3.993\times 10^{-4}\frac{1}{\lceil(1-ß)}\varepsilon^{-ß} \end{pmatrix}. \quad (29)$$

Using $\varepsilon_0 = 0.03$, we have approximated our desired real root of 0.06.

The numerical findings are given in Tables 4 and 5. Tables 4 and 5 clearly indicate that, for a range of values of ß, $MK^{ß}$ performs better than $CK^{ß}$ in terms of efficiency, computing time, number of iterations and residual error.

Example 4.2. Chemical Engineering[41]
The acidity of a MgOH solution in HCl is determined for a $[H_3O^+]$ by

$$\frac{3.64 \times 10^{-11}}{[H_3O^+]} = [H_3O^+] + 3.6 \times 10^{-4}, \quad (30)$$

where HCl is hydrochloric acid and $[H_3O^+]$ is the concentration of hydronium ions. By assuming $\varepsilon = 10^4[H_3O^+]$, we obtain the following nonlinear model, i.e.

$$f(\varepsilon) = \varepsilon^3 + 3.6\varepsilon^2 - 36.4. \quad (31)$$

Exact solution of (31) is $-3 \pm 2.3i$, 2.4. We have approximate our desired real root 2.4 with $\varepsilon_0 = 2.3$.

$$\left[{}_{\breve{c}}\eth^{ß}_{ß_1}\right] f(\varepsilon) = \begin{pmatrix} \frac{\lceil(4)}{\lceil(4-ß)}\varepsilon^{3-ß} + 3.6\frac{\lceil(3)}{\lceil(3-ß)}\varepsilon^{2-ß} \\ -36.4\frac{1}{\lceil(1-ß)}\varepsilon^{-ß} \end{pmatrix}. \quad (32)$$

The numerical findings are given in Tables 6 and 7. Tables 6 and 7 clearly indicate that, for a range of values of ß, $MK^{ß}$ performs better than $CK^{ß}$ in terms of efficiency, computing time, number of iterations and residual error.

Table 4 Fractional Iterative Algorithms Numerical Results for Engineering Application 1.

$CK^{ß}$ method-numerical outcomes				
Approximate roots: $\xi_3 = 0.062377581513$				
ß	**n**	**$\lvert\varepsilon_{i+1} - \varepsilon_i\rvert$**	**$\lvert f(\varepsilon_i)\rvert$**	**CPU**
0.10	36	0.23×10^{-3}	1.17×10^{-4}	0.0778
0.30	24	1.45×10^{-4}	2.14×10^{-4}	0.0762
0.50	22	0.67×10^{-4}	5.07×10^{-5}	0.0785
0.60	12	70.56×10^{-4}	4.94×10^{-5}	0.0562
0.70	10	5.45×10^{-6}	0.77×10^{-7}	0.0332
0.80	08	1.34×10^{-7}	4.69×10^{-9}	0.0983
0.87	08	0.65×10^{-9}	0.7×10^{-10}	0.0454
0.90	07	0.34×10^{-10}	1.43×10^{-13}	0.0082
1.00	06	2.67×10^{-53}	2.74×10^{-99}	0.0013

Table 5 Fractional Iterative Algorithms Numerical Results for Engineering Application 1.

$MK^{ß}$ method-numerical outcomes				
Approximate roots: $\xi_3 = 0.062377581513749505987$				
ß	**n**	**$\lvert\varepsilon_{i+1} - \varepsilon_i\rvert$**	**$\lvert f(\varepsilon_i)\rvert$**	**CPU**
0.10	36	5.7×10^{-2}	0.54×10^{-3}	0.0778
0.30	24	0.7×10^{-2}	0.12×10^{-3}	0.0452
0.50	22	3.47×10^{-3}	1.65×10^{-3}	0.0345
0.60	12	6.54×10^{-3}	3.43×10^{-5}	0.0712
0.70	10	6.76×10^{-4}	0.33×10^{-6}	0.0232
0.80	08	0.88×10^{-5}	5.65×10^{-8}	0.0013
0.87	08	1.23×10^{-7}	4.57×10^{-8}	0.0674
0.90	07	4.45×10^{-9}	9.72×10^{-11}	0.0022
1.00	06	5.23×10^{-62}	8.99×10^{-106}	0.0019

Table 6 Fractional Iterative Algorithms Numerical Results for Engineering Application 2.

$CK^{ß}$ method				
Approximate roots: $\xi_3 = 2.452379213194619124$				
ß	**n**	**$\lvert\varepsilon_{i+1} - \varepsilon_i\rvert$**	**$\lvert f(\varepsilon_i)\rvert$**	**CPU**
0.10	36	9.21×10^{-1}	9.7×10^{-2}	0.0578
0.30	24	7.37×10^{-1}	8.14×10^{-2}	0.0412
0.50	22	1.32×10^{-1}	4.37×10^{-2}	0.0345
0.60	12	0.93×10^{-2}	6.14×10^{-3}	0.0212
0.70	10	2.72×10^{-3}	3.01×10^{-4}	0.0222
0.80	08	1.83×10^{-4}	9.90×10^{-4}	0.0113
0.87	08	0.73×10^{-9}	6.08×10^{-14}	0.0124
0.90	07	4.44×10^{-13}	1.34×10^{-17}	0.0012
1.00	06	5.74×10^{-53}	2.47×10^{-92}	0.0010

Table 7 Fractional Iterative Algorithms Numerical Results for Engineering Application 2.

MK^ß method				
Approximate roots: $\xi_3 = 2.452379213194619124$				
ß	n	$\lvert\varepsilon_{i+1}-\varepsilon_i\rvert$	$\lvert f(\varepsilon_i)\rvert$	CPU
0.10	36	1.71451	1.4×10^{-1}	0.0698
0.30	24	8.57×10^{-1}	1.51×10^{-1}	0.0702
0.50	22	6.65×10^{-1}	0.51×10^{-1}	0.0345
0.60	12	8.47×10^{-2}	3.53×10^{-2}	0.0210
0.70	10	4.37×10^{-2}	1.75×10^{-2}	0.0232
0.80	08	9.91×10^{-3}	8.55×10^{-3}	0.0143
0.87	08	6.77×10^{-4}	5.75×10^{-5}	0.0154
0.90	07	5.57×10^{-27}	3.72×10^{-49}	0.0022
1.00	06	9.57×10^{-97}	4.72×10^{-401}	0.0019

Example 4.3. Civil Engineering Application[42]

The nonlinear equation for the resulting elastic curve is

$$f(\varepsilon)=\frac{\omega_o}{120EIL}(-\varepsilon^5+2L^2\varepsilon^3-L^4\varepsilon). \tag{33}$$

Taking $f'(\varepsilon)=0$, to determine the point of maximum deflection i.e.

$$\frac{\omega_o}{120EIL}(-5z^4+6L^2z^2-L^4)=0,$$

$$f(\varepsilon)=\frac{\omega_o}{120EIL}(-5\varepsilon^4+6L^2\varepsilon^2-L^4). \tag{34}$$

The Caputo-type derivative of (34) is

$$\left[{}_{\check{c}}\partial^{ß}_{ß_1}\right]f(\varepsilon)=\frac{\omega_o}{120EIL}\begin{pmatrix}-5\frac{\lceil(5)}{\lceil(5-ß)}\varepsilon^{4-ß}\\ +6L^2\frac{\lceil(3)}{\lceil(3-ß)}\varepsilon^{2-ß}\\ -L^4\frac{1}{\lceil(1-ß)}\varepsilon^{-ß}\end{pmatrix}. \tag{35}$$

Then, substitute this value in (34) to determine the value of maximum deflection. Use the following values in computation $L=600$ čm, $E=50{,}000$ KN/čm^3, $I=30{,}000$ čm^4 and $\omega_o=2.5$ KN/čm. Equation (34) has four exact roots $\xi_1=-599.999$, $\xi_2=-268.328$, $\xi_3=268.328$, $\xi_4=599.999$ are shown in the figure. Using $\varepsilon_0=-575$, we have approximated our desired real root of 2.4.

The numerical findings are given in Tables 8 and 9. Tables 8 and 9 clearly indicate that, for a range of values of ß, MK^ß performs better than CK^ß in terms of efficiency, computing time, number of iterations and residual error.

Table 8 Fractional Iterative Algorithms Numerical Results for Engineering Application 3.

CK^ß method				
Approximate roots: $\xi_1 = -599.999$				
ß	n	$\lvert\varepsilon_{i+1}-\varepsilon_i\rvert$	$\lvert f(\varepsilon_i)\rvert$	CPU
0.10	36	0.61×10^{-3}	9.27×10^{-4}	2.1722
0.30	24	1.47×10^{-4}	8.34×10^{-4}	3.4222
0.50	22	0.34×10^{-4}	5.47×10^{-5}	2.9371
0.60	12	0.62×10^{-4}	6.34×10^{-5}	1.2815
0.70	10	5.77×10^{-6}	3.67×10^{-7}	1.0125
0.80	08	0.72×10^{-7}	0.79×10^{-9}	0.3652
0.87	08	0.97×10^{-9}	0.77×10^{-10}	0.2145
0.90	07	6.69×10^{-10}	1.36×10^{-13}	0.0412
1.00	06	4.99×10^{-33}	0.95×10^{-39}	0.0213

Table 9 Fractional Iterative Algorithms Numerical Results for Engineering Application 3.

MK^ß method				
Approximate roots: $\xi_1 = -599.999$				
ß	n	$\lvert\varepsilon_{i+1}-\varepsilon_i\rvert$	$\lvert f(\varepsilon_i)\rvert$	CPU
0.10	36	6.33×10^{-2}	2.91×10^{-3}	2.9711
0.30	24	3.47×10^{-2}	1.90×10^{-3}	3.5622
0.50	22	7.44×10^{-3}	0.91×10^{-3}	2.8710
0.60	12	9.74×10^{-3}	8.17×10^{-5}	1.4321
0.70	10	9.31×10^{-4}	9.92×10^{-6}	1.0115
0.80	08	3.69×10^{-5}	4.72×10^{-8}	0.3142
0.87	08	2.77×10^{-7}	5.12×10^{-8}	00.124
0.90	07	4.47×10^{-39}	6.27×10^{-101}	0.0542
1.00	06	6.66×10^{-29}	7.72×10^{-306}	0.0049

4.1. Results and Discussion

The fractal analysis and numerical results of the iterative schemes are shown in Tables 1–9, which illustrate how effective fractional iterative methods are at resolving nonlinear engineering problems. There are significant differences in the fractional-order techniques' convergence rate, computational accuracy, stability and efficiency when compared.

- The proposed method MK^ß outperforms existing approach CK^ß to fractal generation. It is better suited for real-world applications since it achieves greater precision with less elapsed time (Tables 1–3). The outcomes validate that our approach offers a more sophisticated computational framework for fractals investigations (Figs. 1–3).
- In numerical calculations (Tables 3–9), our method significantly reduces the residual error

compared to existing methods. The results show that the error decreases consistently as the fractional parameter increases, validating the correctness and reliability of the suggested approach.

- The findings suggest that our scheme $MK^{ß}$ requires less CPU time to obtain convergence as compared to $CK^{ß}$. This advancement is essential for effectively resolving complicated nonlinear equations, particularly in engineering applications and large-scale simulations.
- The proposed fractional iterative method is more stable and has a higher rate of convergence (Tables 5, 7 and 8), particularly for larger fractional parameter values. This ensures that the approach accurately approximates answers while remaining numerically stable across varied initial starting values.
- Our approach is reliable and robust because it remains consistent throughout a range of fractional values. The results are consistent across multiple test settings, proving its ability to handle nonlinear problems.
- The method's accuracy and efficiency are largely determined by the fractional parameter ß. As ß increases from 0 to 1, the proposed approach's accuracy, stability and efficiency all improve (Tables 4–9). Our approach $MK^{ß}$ is a better alternative for nonlinear engineering problems because it performs better than existing approaches in all respects when ß is close to 1.

Thus, this study investigates how these techniques might be applied to real-world engineering and scientific problems, illustrating their utility in comparison to existing methods. Using both theoretical investigations and simulations, the proposed methods aim to provide an exceptional and reliable basis for solving complex nonlinear problems.

5. CONCLUSION

The Caputo-type fractional iterative approach with convergence order 2ß+1 was developed for solving nonlinear equations. Convergence analysis is thoroughly investigated, and numerical findings demonstrate that the computational order of convergence verifies the theoretical convergence order. In order to demonstrate the stability of the recently developed scheme, its fractal behavior is thoroughly investigated in comparison to the existing technique as shown in Tables 1–3. In compliance with Tables 4–9, the numerical outcomes of the engineering applications show that $MK^{ß}$ performs better than $CK^{ß}$ in terms of iterations, residual error and computational order of convergence for different fractional parameter values ß.

Future research will focus on improving fractional iterative algorithms by creating higher-order schemes capable of solving complex nonlinear models, such as those used in epidemiology and engineering. In order to increase computing efficiency and accuracy, the Caputo–Fabrizio fractional derivative — which is renowned for its non-singular kernel properties — will be used. In addition, compatibility with high-performance computing techniques could speed up the solution of large-scale fractional problems.

ACKNOWLEDGMENTS

The publication of this paper was funded by the University of Oradea, Romania.

ORCID

Mudassir Shams
https://orcid.org/0000-0002-2980-5801
Nasreen Kausar
https://orcid.org/0000-0002-8659-0747
Hossein Jafari
https://orcid.org/0000-0001-6807-6675
Georgia Irina Oros
https://orcid.org/0000-0003-2902-4455

REFERENCES

1. M. M. Dzherbashyan and A. B. Nersesian, About application of some integro-differential operators, *Dokl. Akad. Nauk* **121**(2) (1958) 210–213.
2. K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations* (John Wiley & Sons, Inc., New York, 1993).
3. K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Application of Differentiation and Integration to Arbitrary Order* (Academic Press, New York, 1974).
4. R. P. Agarwal, V. Lakshmikantham and J. J. Nieto, On the concept of solution for fractional differential equations with uncertainty, *Nonlinear Anal. Theory Methods Appl.* **72**(6) (2010) 2859–2862.
5. D. R. Anderson and D. J. Ulness, Newly defined conformable derivatives, *Adv. Dyn. Syst. Appl.* **10**(2) (2015) 109–137.
6. S. A. El-Wakil and E. M. Abulwafa, Formulation and solution of space-time fractional Boussinesq equation, *Nonlinear Dyn.* **80**(1–2) (2015) 167–175.

7. D. Zhao and M. Luo, General conformable fractional derivative and its physical interpretation, *Calcolo* **54** (2017) 903–917.
8. V. J. Ervin and J. P. Roop, Variational solution of fractional advection dispersion equations on bounded domains in $\mathbb{R}^d$, *Numer. Methods Partial Differ. Equ.* **23**(2) (2007) 256–281.
9. A. M. Mathai and H. J. Haubold, *Fractional and Multivariable Calculus: Model Building and Optimization Problems*, Springer Optimization and Its Applications, Vol. 122 (Springer, Berlin, 2017).
10. T. M. Atanackovic, S. Pilipovic, B. Stankovic and D. Zorica, *Fractional Calculus with Applications in Mechanics: Wave Propagation, Impact and Variational Principles* (Wiley, London, 2014).
11. M. Saravi and S. Saravi, A new look at one of the Newton-like methods for solving non-linear equations, *Int. J. Numer. Methods Appl.* **5**(1) (2011) 31–43.
12. M. Shams, N. Rafique, N. Kausar, P. Agrawal, C. Park and N. A. Mir, On iterative technique for estimating all roots of non-linear equations and its system with applications in differential equations, *Adv. Differ. Equ.* **2021** (2021) 480.
13. K. Pavani, K. Raghavendar and K. Aruna, Solitary wave solutions of the time fractional Benjamin Bona Mahony Burger equation, *Sci. Rep.* **14**(1) (2024) 14596.
14. Y. S. Gasimov, A. Nachaoui and A. A. Niftiyev, Non-linear eigenvalue problems for p-Laplacian with variable domain, *Optim. Lett.* **4** (2010) 67–84.
15. H. Jafari, R. M. Ganji, D. D. Ganji, Z. Hammouch and Y. S. Gasimov, A novel numerical method for solving fuzzy variable-order differential equations with Mittag-Leffler kernels, *Fractals* **31**(4) (2023) 2340063.
16. G. Ramasekhar, J. Shaik, S. R. Reddisekhar Reddy, A. Divya, M. Jawad and B. A. Ali Yousif, Numerical investigation of Casson fluid flow performance of blood containing gold and Fe_3O_4 nanofluid injected into a stenotic artery, *Numer. Heat Transf. A* **2024** (2024) 1–17.
17. K. Pavani, K. Raghavendar and K. Aruna, An efficient technique to study time fractional extended Fisher–Kolmogorov equation, *Contemp. Math.* **5**(2) (2024) 1137–1153.
18. G. Ramasekhar, S. Jakeer, S. R. R. Reddy, S. Alkarni and N. A. Shah, Biomedical importance of Casson nanofluid flow with silver and Fe_2O_3 nanoparticles delivered into a stenotic artery: Numerical study, *AIMS Math.* **9**(8) (2024) 23142–23157.
19. J. He, G. Moatimid and M. Zekry, Forced nonlinear oscillator in a fractal space, *Facta Univ. Ser. Mech. Eng.* **20**(1) (2022) 001–020.
20. K. Pavani, K. Raghavendar and K. Aruna, Soliton solutions of the time-fractional Sharma–Tasso–Olver equations arise in nonlinear optics, *Opt. Quantum Electron.* **56**(5) (2024) 748.
21. H. Jafari, R. M. Ganji, D. D. Ganji, Z. Hammouch and Y. S. Gasimov, A novel numerical method for solving fuzzy variable-order differential equations with Mittag-Leffler kernels, *Fractals* **31**(4) (2023) 2340063.
22. G. Ramasekhar and P. Bala Anki Reddy, Entropy generation on Darcy–Forchheimer flow of copper-aluminium oxide/water hybrid nanofluid over a rotating disk: Semi-analytical and numerical approaches, *Sci. Iran.* **30**(6) (2023) 2245–2259.
23. H. Jafari, H. Tajadodi, S. R. M. Khatir and V. T. Nguyen, Fractional variational problem involving indefinite integrals and nonsingular kernels, *Fractals* **31**(4) (2023) 2340048.
24. H. Ma, Simplified Hamiltonian-based frequency-amplitude formulation for nonlinear vibration systems, *Facta Univ. Ser. Mech. Eng.* **20**(2) (2022) 445–455.
25. K. Pavani and K. Raghavendar, A novel method to study time fractional coupled systems of shallow water equations arising in ocean engineering, *AIMS Math.* **9**(1) (2024) 542–564.
26. Y. T. Zuo, F. F. Luo and S. L. Zeng, Gecko-inspired fractal buffer for passenger elevator, *Facta Univ. Ser. Mech. Eng.* **22**(4) (2024) 633–642.
27. A. F. Torres-Hernandez, A. F. Brambila-Paz, U. Iturrarán-Viveros and R. Caballero-Cruz, Fractional Newton–Raphson method accelerated with Aitken's method, *Axioms* **10**(2) (2021) 47.
28. A. Akgül, A. Cordero and J. R. Torregrosa, A fractional Newton method with 2αth-order of convergence and its stability, *Appl. Math. Lett.* **98** (2019) 344–351.
29. F. Cajori, Historical note on the Newton–Raphson method of approximation, *Am. Math. Mon.* **18**(2) (1911) 29–32.
30. P. Kumar and O. P. Agrawal, An approximate method for numerical solution of fractional differential equations, *Signal Process.* **86**(10) (2006) 2602–2610.
31. K. Gdawiec, W. Kotarski and A. Lisowska, Newton's method with fractional derivatives and various iteration processes via visual analysis, *Numer. Algorithms* **86**(3) (2021) 953–1010.
32. M. P. Ineh, E. P. Akpan and H. A. Nabwey, A novel approach to Lyapunov stability of Caputo fractional dynamic equations on time scale using a new generalized derivative, *AIMS Math.* **9**(12) (2024) 34406–34434.
33. Z. M. Odibat and N. T. Shawagfeh, Generalized Taylor's formula, *Appl. Math. Comput.* **186** (2007) 286–293.

34. I. K. Argyros and S. Hilout, *Computational Methods in Nonlinear Analysis* (World Scientific Publishing Company, New Jersey, 2013).
35. G. Candelario, A. Cordero and J. R. Torregrosa, Multipoint fractional iterative methods with $(2\alpha + 1)$th-order of convergence for solving nonlinear problems, *Mathematics* **8**(3) (2020) 452.
36. M. Shams, N. Kausar, P. Agarwal, S. Jain, M. A. Salman and M. A. Shah, On family of the Caputo-type fractional numerical scheme for solving polynomial equations, *Appl. Math. Sci. Eng.* **31**(1) (2023) 2181959.
37. A. M. Ostrowski, *Solution of Equations in Euclidean and Banach Space*, 3rd edn. (Academic Press, New York, 1973).
38. M. Saqib, D. Ahmad, A. N. Al-Kenani and T. Allahviranloo, Fourth- and fifth-order iterative schemes for nonlinear equations in coupled systems: A novel Adomian decomposition approach, *Alex. Eng. J.* **74** (2023) 751–760.
39. M. Kansal and H. Sharma, Analysis of optimal iterative methods from a dynamical point of view by studying their stability properties, *J. Math. Chem.* **62**(1) (2024) 198–221.
40. M. Saravi and S. Saravi, A new look at one of the Newton-like methods for solving non-linear equations, *Int. J. Numer. Methods Appl.* **5**(1) (2011) 31–43.
41. C. Steven Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 6th edn. (McGraw-Hill, New York, 2010).
42. A. Naseem, M. A. Rehman and T. Abdeljawad, Computational methods for non-linear equations with some real-world applications and their graphical analysis, *Intell. Autom. Soft Comput.* **30**(3) (2021) 1–15.