Research Article

Maroua Amel Boubekeur, Salah Boulaaras*, and Seda Igret Araz*

Successive midpoint method for fractional differential equations with nonlocal kernels: Error analysis, stability, and applications

https://doi.org/10.1515/phys-2025-0203
received May 19, 2025; accepted July 28, 2025

Abstract: This research introduces a numerical approach that employs the midpoint technique successively to solve differential equations that contain Atangana–Baleanu and Caputo fractional derivatives. The stability and consistency of the proposed approach has been demonstrated through the application of the technique suggested for solving the perturbed problem. Furthermore, the error analysis has been carried out for a nonlinear equation with Caputo derivatives. The effectiveness of this method is illustrated by examining various examples incorporating nonlocal kernels. Finally, by the use of an example, the well-known Fitzhugh–Nagumo model with Atangana–Baleanu fractional derivatives is simulated by using the proposed scheme. The method exhibits improved accuracy, consistency, and stability compared to the classical midpoint method, with particularly good performance for the Atangana–Baleanu and Caputo cases due to a specially designed predictor component. The novelty of the work lies in extending the method to fractional operators with non-local and non-singular kernels, offering a more robust approach than existing techniques in the literature.

Keywords: successive midpoint, stability, convergence, Fitzhugh–Nagumo model, fractional derivatives

*** Corresponding author: Salah Boulaaras,** Department of Mathematics, College of Science, Qassim University, Buraydah 52571, Saudi Arabia, e-mail: s.boularas@qu.edu.sa

*** Corresponding author: Seda Igret Araz**, Department of Mathematic Education, Siirt University, Siirt, Turkey, e-mail: sedaaraz@siirt.edu.tr

Maroua Amel Boubekeur: Department of Mathematics and Computer Science, University of Mostaganem, Mostganem, Algeria, e-mail: maroua.boubekeur.etu@univ-mosta.dz

1 Introduction and preliminaries

Numerical methods are mathematical techniques designed to find approximate solutions to complex problems that cannot be solved analytically. Given that real-world challenges in science, engineering, and finance often involve complicated equations, these methods serve as vital tools for modeling and addressing such problems. The evolution of numerical methods has significantly impacted various fields, enabling scientists and engineers to effectively simulate physical processes, analyze data, and optimize systems. This category of methods includes a diverse array of techniques such as interpolation, numerical integration, differentiation, and the resolution of ordinary and partial differential equations. Improvements in algorithms and computing technology have enhanced the accuracy and efficiency of numerical solutions, making them essential in both theoretical and practical contexts. Some of the methods that are fundamental to numerical analysis and are widely used in various scientific and engineering disciplines to solve practical problems are Euler methods [1], Runge–Kutta methods [2], midpoint methods [3], finite difference methods, interpolation techniques such as and Lagrange [4] and Newton interpolation [5–7], numerical integrations like the trapezoidal rule and Simpson's rule, spectral methods [8], Kudryashov methods [9], and the collocation method [10].

Recently, a numerical technique called the successive midpoint method has been introduced in the literature [11]. While this method has been developed for classical, Caputo Fabrizio, global, and fractal derivatives, but its derivation for Atangana–Baleanu and Caputo fractional derivatives is not available in the literature [12–14]. This study develops a numerical method that utilizes the midpoint method iteratively to solve differential equations with Atangana–Baleanu and Caputo fractional derivatives. Then, we considered the Fitzhugh–Nagumo equation [15,16] which provides a simplified framework for the study of complex biological systems. This equation is extremely important for medical applications such as understanding the behavior of excitable cells, designing drugs that regulate heart rhythm, or studying neurological disorders such as epilepsy. By solving this equation, researchers can gain insight into emergent phenomena observed in biological networks. As a practical implementation of the proposed technique, we have established the scheme for the numerical solution of the Fitzhugh–Nagumo equation with Atangana–Baleanu fractional derivatives and carried out the numerical simulations for such model.

We shall present the definitions of the fractional derivatives that will be used here before proceeding with the derivation of the numerical method.

The definition of the Riemann–Liouville fractional derivative is represented by

$$ {}_{t_0}^{RL}D_t u(t) = \frac{1}{\Gamma(1-\alpha)}\frac{\mathrm{d}}{\mathrm{d}t}\int_{t_0}^{t} u(\tau)(t-\tau)^{-\alpha}\mathrm{d}\tau, \tag{1} $$

where the function $u(\cdot)$ is continuous and $0 < \alpha < 1$. The associated integral is given as

$$ {}_{t_0}^{RL}I_t u(t) = \frac{1}{\Gamma(\alpha)}\int_{t_0}^{t} u(\tau)(t-\tau)^{\alpha-1}\mathrm{d}\tau. \tag{2} $$

The definition of the Caputo fractional derivative is defined by

$$ {}_{t_0}^{C}D_t u(t) = \frac{1}{\Gamma(1-\alpha)}\int_{t_0}^{t} u'(\tau)(t-\tau)^{-\alpha}\mathrm{d}\tau, \tag{3} $$

where the function $u(\cdot)$ is differentiable and $0 < \alpha < 1$. The definition of the Atangana–Baleanu fractional derivative is given by the following:

$$ {}_{t_0}^{AB}D_t u(t) = \frac{1}{1-\alpha}\int_{t_0}^{t} u'(\tau)E_\alpha\left[-\frac{\alpha}{1-\alpha}(t-\tau)^\alpha\right]\mathrm{d}\tau, \tag{4} $$

where the function $u(\cdot)$ is differentiable and $0 < \alpha < 1$. The associated integral is stated by

$$ {}_{t_0}^{AB}I_t u(t) = (1-\alpha)u(t) + \frac{\alpha}{\Gamma(\alpha)}\int_{t_0}^{t} u(\tau)(t-\tau)^{\alpha-1}\mathrm{d}\tau. \tag{5} $$

The study is organized as follows: In Section 2, we present the derivation of a successive midpoint formula for a general Cauchy problem with the Atangana–Baleanu fractional derivative. In Section 3, we derive the successive midpoint formula for the Caputo case. Also, in these sections, the error of the exact and approximate solutions of the relevant problem is investigated, and the results are compared with the midpoint and L1 methods to demonstrate the accuracy of the proposed scheme. In Section 4, we demonstrate the stability and consistency of the associated method and present error analysis for a general initial value problem with classical and Caputo fractional derivatives. In Section 5, applications of the suggested method to the Fitzhugh–Nagumo model with Atangana–Baleanu fractional derivatives are presented and the numerical simulations are performed for such model.

2 Successive midpoint formula for a general Cauchy problem with Atangana–Baleanu fractional derivatives

In this section, we apply the successive midpoint formula introduced in the literature [11] to a general Cauchy problem with Atangana–Baleanu fractional derivatives

$$ \begin{aligned} {}_0^{AB}D_t^\alpha u(t) &= \gamma(t, u(t)), \\ u(0) &= u_0. \end{aligned} \tag{6} $$

Applying the Atangana–Baleanu integral yields

$$ \begin{aligned} u(t) - u(0) = {} & (1-\alpha)\gamma(t, u(t)) \\ & + \frac{\alpha}{\Gamma(\alpha)}\int_0^t \gamma(\tau, u(\tau))(t-\tau)^{\alpha-1}\mathrm{d}\tau. \end{aligned} \tag{7} $$

At the point $t = t_{n+1}$, we write

$$ \begin{aligned} u(t_{n+1}) = {} & u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\ & + \frac{\alpha}{\Gamma(\alpha)}\int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau. \end{aligned} \tag{8} $$

Employing the midpoint method consecutively [11], we obtain

$$ \begin{aligned} u(t_{n+1}) = {} & u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\ & + \frac{\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\gamma\left(\left(t_j + \frac{h}{2}, u\left(t_j + \frac{h}{2}\right)\right)\right) \\ & \times \int_{t_j}^{t_{j+1}} (t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau \\ = {} & u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\ & + \frac{h^\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\gamma\left(\left(t_j + \frac{h}{2}, u\left(t_j + \frac{h}{2}\right)\right)\right) \\ & \times ((n-j+1)^\alpha - (n-j)^\alpha). \end{aligned} \tag{9} $$

Dividing the interval second time, we have

$$
\begin{aligned}
u(t_{n+1}) &= u(0) + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j}^{t_j+\frac{h}{2}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\quad + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\
&\quad + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j+\frac{h}{2}}^{t_{j+1}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau \\
&= u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{h}{4}, u\left(t_j + \frac{h}{4}\right)\right)\right) \\
&\quad \times \left((n-j+1)^\alpha - \left(n-j+\left(1-\frac{h}{2}\right)\right)^\alpha\right) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{3h}{4}, u\left(t_j + \frac{h}{4}\right)\right)\right)\left(\left(n-j+\left(1-\frac{h}{2}\right)\right)^\alpha - (n-j)^\alpha\right).
\end{aligned}
\tag{10}
$$

Dividing the third time, we obtain

$$
\begin{aligned}
u(t_{n+1}) &= u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\
&\quad + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j}^{t_j+\frac{h}{4}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\quad + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j+\frac{h}{4}}^{t_j+\frac{h}{2}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\quad + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j+\frac{h}{2}}^{t_j+\frac{3h}{4}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\quad + \frac{\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j+\frac{3h}{4}}^{t_{j+1}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} \mathrm{d}\tau
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
&= u(0) + (1-\alpha)\gamma(t_{n+1}, u(t_{n+1})) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{h}{8}, u\left(t_j + \frac{h}{8}\right)\right)\right) \\
&\quad \times \left((n-j+1)^\alpha - \left(n-j+\left(1-\frac{h}{4}\right)\right)^\alpha\right) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{3h}{8}, u\left(t_j + \frac{h}{8}\right)\right)\right)\left(\begin{array}{c}\left(n-j+\left(1-\frac{h}{4}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{h}{2}\right)\right)^\alpha\end{array}\right) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{5h}{8}, u\left(t_j + \frac{5h}{8}\right)\right)\right)\left(\begin{array}{c}\left(n-j+\left(1-\frac{h}{2}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{3h}{4}\right)\right)^\alpha\end{array}\right) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma\left(\left(t_j + \frac{7h}{8}, u\left(t_j + \frac{7h}{8}\right)\right)\right)\left(\begin{array}{c}\left(n-j+\left(1-\frac{3h}{4}\right)\right)^\alpha \\ -(n-j)^\alpha\end{array}\right).
\end{aligned}
$$

Using the idea of the midpoint consecutively and adding the predictor term, we obtain the following numerical scheme:

$$
\begin{aligned}
u_{n+1} &= u(0) + (1-\alpha)\gamma(t_{n+1}, \tilde{u}_{n+1}) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \left[\begin{array}{c} \sum\limits_{m=0}^{2^{k-1}-1} \gamma\left(t_j + \frac{(2m+1)h}{2^k}, u\left(t_j + \frac{(2m+1)h}{2^k}\right)\right) \\ \times\left(\begin{array}{c}\left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha\end{array}\right)\end{array}\right].
\end{aligned}
\tag{12}
$$

It is worth noting that we need to calculate the predictor term. The predictor term is evaluated as

$$
\begin{aligned}
\tilde{u}_{n+1} &= u(0) + (1-\alpha)\gamma(t_n, u_n) \\
&\quad + \frac{h^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} \gamma((t_j, u_j))((n-j+1)^\alpha - (n-j)^\alpha).
\end{aligned}
\tag{13}
$$

Example 1. We consider the following initial value problem:

$$
\begin{aligned}
{}_{0}^{AB}D_t^\alpha u(t) &= t^2 \\
u(0) &= 0,
\end{aligned}
\tag{14}
$$

where the exact solution is

$$
u(t) = (1-\alpha)t^2 + \frac{\alpha\Gamma(3)}{\Gamma(\alpha+3)} t^{\alpha+2}. \tag{15}
$$

Table 1: Error of the function $u(t)$ for the suggested method

h	MM for $\alpha = 1, k = 1$	L1 method for $\alpha = 1$	Suggested method for $\alpha = 1, k = 10$	CPU time (s)
0.1	8.3333×10^{-4}	0.0078	3.1789×10^{-9}	0.517
0.05	2.0833×10^{-4}	0.0020	7.9473×10^{-10}	0.956
0.01	8.3333×10^{-6}	8.2833×10^{-5}	3.1790×10^{-11}	8.601
h	**MM for $\alpha = 0.9$**	**L1 method for $\alpha = 0.9$**	**Suggested method for $\alpha = 0.9, k = 11$**	**CPU time (s)**
0.1	0.0016	0.0265	3.8039×10^{-9}	4.326
0.05	4.4283×10^{-4}	0.0117	1.0326×10^{-9}	16.841
0.01	2.2208×10^{-5}	0.0021	4.9815×10^{-11}	494.9-45
h	**MM for $\alpha = 0.8$**	**L1 method for $\alpha = 0.8$**	**Suggested method for $\alpha = 0.8, k = 12$**	**CPU time (s)**
0.1	0.0026	0.0450	3.4677×10^{-9}	6.083
0.05	7.7191×10^{-4}	0.0213	1.0022×10^{-9}	22.375
0.01	4.5448×10^{-5}	0.0041	5.5959×10^{-11}	536.00

Table 1 presents the numerical errors of the function $u(t)$ obtained using the suggested method for various values of the fractional and classical orders. The results are compared with those of the midpoint (MM) [3] and L1 methods, where in the latter, the function $\gamma(t, u(t))$ is approximated using the first-order Lagrange interpolation [4]. Also, the CPU time (for an Intel Core 9, 2.7 GHz) for each simulation was recorded and presented in Table 1 to evaluate the computational efficiency of the proposed method.

As seen in Table 1, the CPU time required by the suggested method increases significantly as the fractional order α decreases and the parameter k increases. This trend is particularly evident for smaller time step sizes h, where the CPU time grows rapidly – reaching over 536 s when $\alpha = 0.8$, $k = 12$, and $h = 0.01$. The reason for this behavior lies in the nonlocal nature of fractional derivatives: as α decreases, the memory effect becomes more dominant, requiring more historical data to be processed at each time step. Additionally, increasing k introduces further subdivisions and intermediate calculations, contributing to higher computational complexity. In contrast, the midpoint [3] and L1 methods [4] show relatively stable and much lower CPU times, even at smaller h values. However, this comes at the cost of reduced accuracy, as reflected in their higher error values compared to the suggested method. In Figure 1, the numerical simulation is provided for the above problem with the Atangana–Baleanu fractional derivative.

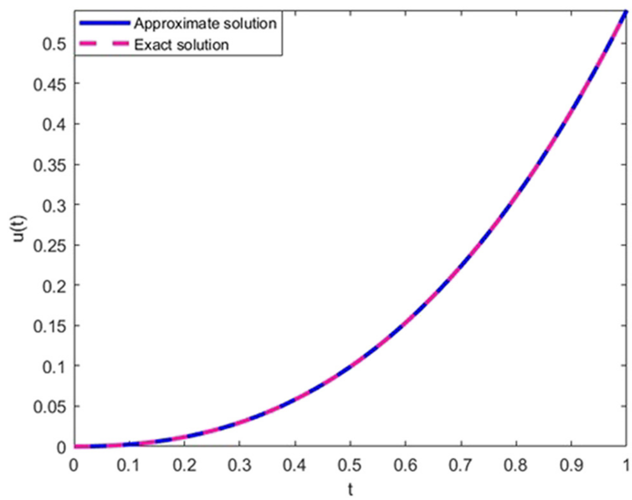

Figure 1: Numerical simulation for the Cauchy problem with the Atangana–Baleanu fractional derivative for $\alpha = 0.8$, $k = 11$.

3 Successive midpoint method for a general Cauchy problem with Caputo fractional derivatives

In this section, we apply the successive midpoint formula introduced in the literature [11] to a general Cauchy problem with Caputo fractional derivative

$$\begin{aligned} {}_0^C D_t^\alpha u(t) &= \gamma(t, u(t)), \\ u(0) &= u_0. \end{aligned} \tag{16}$$

Applying the associated integral yields

$$u(t) - u(0) = \frac{1}{\Gamma(\alpha)} \int_0^t \gamma(\tau, u(\tau))(t-\tau)^{\alpha-1} \mathrm{d}\tau. \tag{17}$$

At the point $t = t_{n+1}$, we write

$$u(t_{n+1}) = u_0 + \frac{1}{\Gamma(\alpha)} \int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1} \mathrm{d}\tau. \tag{18}$$

The above is converted to

$$u(t_{n+1}) = u_0 + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1} \mathrm{d}\tau. \tag{19}$$

Using the idea of the successive midpoint, we obtain the following numerical scheme:

$$u_{n+1} = u_0 + \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^{n} \left[\sum_{m=0}^{2^{k-1}-1} \gamma\left(t_j + \frac{(2m+1)h}{2^k}, u\left(t_j + \frac{(2m+1)h}{2^k}\right)\right) \times \left(\left(n - j + \left(1 - \frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n - j + \left(1 - \frac{m+1}{2^{k-1}}\right)\right)^\alpha \right) \right]. \tag{20}$$

Example 2. We consider the following initial value problem:

$$\begin{aligned} {}_0^C D_t^\alpha u(t) &= t^3 + t^2, \\ u(0) &= 0, \end{aligned} \tag{21}$$

where the exact solution is

$$u(t) = \frac{6}{\Gamma(\alpha+4)} t^{\alpha+3} + \frac{2}{\Gamma(\alpha+3)} t^{\alpha+2}. \tag{22}$$

In Table 2, we present the error of the function $u(t)$ by employing the suggested method for different values of fractal order and classical order. The total CPU times of the methods were computed and reported in Table 2 in order to compare the computational performance.

As observed from Table 2, the CPU time required for the proposed method increases significantly as the fractional order α decreases and the associated parameter k increases. This is particularly evident for smaller step sizes h where the computational burden becomes more pronounced. The increased in CPU time is due to the enhanced memory effect as the fractional order α decreases and the increased number of divisions resulting from larger values of k. In contrast, classical methods (midpoint and L1) remain relatively stable in terms of CPU time, but exhibit inferior accuracy compared to the suggested approach. In Figure 2, the numerical simulation is provided for the above problem with the Caputo fractional derivative.

4 Consistency and stability analysis

Consistency of a method refers to its ability to produce results that converge to the exact solution of the ordinary differential equations as the step size (or time increment) approaches zero [5]. In other words, a consistent numerical method ensures that as you refine the discretization (by decreasing the step size), the numerical solution approaches the exact solution of the differential equation [15–24]. In this section, we present the stability and consistency of the suggested numerical scheme. We start with stability, we choose $\tilde{u}_{n+1}$ and $\tilde{u}_{j+\frac{(2m+1)h}{2^k}}$ the perturbed terms of u_{n+1} and $u_{j+\frac{(2m+1)h}{2^k}}$, respectively. We recall that the approximate solution is given as

$$u_{n+1} = u_0 + \frac{h^\alpha}{\Gamma(\alpha+1)}\sum_{j=0}^{n}\left[\sum_{m=0}^{2^{k-1}-1}\gamma\left(t_j+\frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}}\right) \times \left(\left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha\right)\right]. \quad (23)$$

The perturbed equation is

$$u_{n+1}+\tilde{u}_{n+1} = u_0+\tilde{u}_0+\frac{h^\alpha}{\Gamma(\alpha+1)} \times \sum_{j=0}^{n}\left[\sum_{m=0}^{2^{k-1}-1}\gamma\left(t_j+\frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}} + \tilde{u}_{j+\frac{(2m+1)h}{2^k}}\right)\right] \times \left(\left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha\right). \quad (24)$$

Removing the approximate solution from the perturbed equations yields

Table 2: Error of the function $u(t)$ for the suggested method

h	MM for $\alpha = 1, k = 1$	L1 method for $\alpha = 1$	Suggested method for $\alpha = 1, k = 8$	CPU time (s)
0.1	0.0021	0.0189	1.2716×10^{-7}	0.376
0.05	5.2083×10^{-4}	0.0050	3.1789×10^{-8}	0.452
0.01	2.0833×10^{-5}	2.0634×10^{-4}	1.2716×10^{-9}	2.712
h	MM for $\alpha = 0.9$	L1 method for $\alpha = 0.9$	Suggested method for $\alpha = 0.9, k = 9$	CPU time (s)
0.1	0.0044	0.0207	1.4185×10^{-7}	2.337
0.05	0.0012	0.0054	3.8621×10^{-8}	7.374
0.01	6.0988×10^{-5}	2.2211×10^{-4}	1.8742×10^{-9}	250.453
h	MM for $\alpha = 0.8$	L1 method for $\alpha = 0.8$	Suggested method for $\alpha = 0.8, k = 10$	CPU time (s)
0.1	0.0079	0.0228	7.9473×10^{-9}	4.525
0.05	0.0024	0.0059	1.9868×10^{-9}	14.288
0.01	1.4047×10^{-4}	2.3858×10^{-4}	7.9474×10^{-11}	692.56

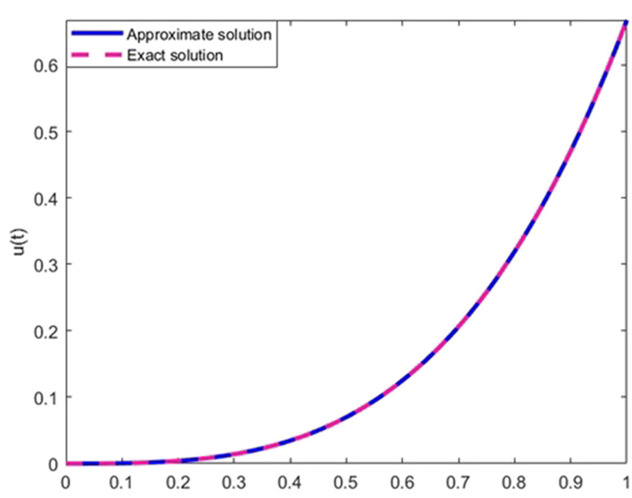

Figure 2: Numerical simulation for the Cauchy problem with the Caputo fractional derivative for $\alpha = 0.8$, $k = 10$.

$$\tilde{u}_{n+1} = \tilde{u}_0 + \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^{n} \left[\sum_{m=0}^{2^{k-1}-1} \left[\begin{array}{c} \gamma\left(t_j + \frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}} + \tilde{u}_{j+\frac{(2m+1)h}{2^k}}\right) \\ -\gamma\left(t_j + \frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}}\right) \end{array} \right] \right] \times \left(\begin{array}{c} \left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha \end{array} \right). \tag{25}$$

Taking the absolute value, we obtain

$$|\tilde{u}_{n+1}| \le |\tilde{u}_0| + \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^{n} \left[\sum_{m=0}^{2^{k-1}-1} \left| \begin{array}{c} \gamma\left(t_j + \frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}} + \tilde{u}_{j+\frac{(2m+1)h}{2^k}}\right) \\ -\gamma\left(t_j + \frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}}\right) \end{array} \right| \right] \times \left(\begin{array}{c} \left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha \end{array} \right). \tag{26}$$

The Lipschitz condition of the function γ with respect to u leads to

$$|\tilde{u}_{n+1}| \le |\tilde{u}_0| + \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^{n} \left[\sum_{m=0}^{2^{k-1}-1} \left| \tilde{u}_{j+\frac{(2m+1)h}{2^k}} \right| \left(\begin{array}{c} \left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha \end{array} \right) \right]. \tag{27}$$

To continue our proof, we need to evaluate

$$\left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha. \tag{28}$$

Employing the mean value theorem, we have

$$\left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha = \alpha(\alpha-1)\xi^{\alpha-1}\frac{1}{2^{k-1}}. \tag{29}$$

By taking $\tilde{z} = \max_{0\le j\le n}\left\{\sum_{m=0}^{2^{k-1}-1}\left|\tilde{u}_{j+\frac{(2m+1)h}{2^k}}\right|\right\}$ then

$$\begin{aligned} |\tilde{u}_{n+1}| &\le |\tilde{u}_0| + \frac{(\alpha-1)h^\alpha}{\Gamma(\alpha)}\tilde{z}\xi^{\alpha-1}\frac{n}{2^{k-1}} \\ &\le c\tilde{z}(n+1), \end{aligned} \tag{30}$$

where

$$c = \frac{(\alpha-1)h^\alpha}{\Gamma(\alpha)2^{k-1}}\xi^{\alpha-1}. \tag{31}$$

This shows that the scheme is stable. We next want to show that

$$\lim_{\Delta t\to 0} |u(t_{n+1}) - u_{n+1}| = 0. \tag{32}$$

Noting that

$$u(t_{n+1}) = u_0 + \frac{1}{\Gamma(\alpha)}\int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1}d\tau \tag{33}$$

and

$$u_{n+1} = u_0 + \frac{h^\alpha}{\Gamma(\alpha+1)}\sum_{j=0}^{n}\left[\begin{array}{c} \sum_{m=0}^{2^{k-1}-1}\gamma\left(t_j + \frac{(2m+1)h}{2^k}, u_{j+\frac{(2m+1)h}{2^k}}\right) \\ \times \left(\begin{array}{c} \left(n-j+\left(1-\frac{m}{2^{k-1}}\right)\right)^\alpha \\ -\left(n-j+\left(1-\frac{m+1}{2^{k-1}}\right)\right)^\alpha \end{array} \right) \end{array} \right]. \tag{34}$$

Taking difference the above equations yields

$$|u(t_{n+1}) - u_{n+1}| = \left| \begin{array}{c} \frac{1}{\Gamma(\alpha)}\int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1}d\tau \\ -\frac{1}{\Gamma(\alpha)}\sum_{j=0}^{n}\int_{t_j}^{t_{j+1}} \gamma(t_j, u_j)(t_{n+1}-\tau)^{\alpha-1}d\tau \end{array} \right| \tag{35}$$

and

$$\begin{aligned} |u(t_{n+1}) - u_{n+1}| &= \frac{1}{\Gamma(\alpha)}\left| \begin{array}{c} \int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1}-\tau)^{\alpha-1}d\tau \\ -\sum_{j=0}^{i}\int_{t_j}^{t_{j+1}} \gamma(t_j, u_j)(t_{n+1}-\tau)^{\alpha-1}d\tau \end{array} \right| \\ &\le \int_0^{t_{n+1}} |\gamma(\tau, u(\tau)) - \gamma(t_j, u_j)|(t_{n+1}-\tau)^{\alpha-1}d\tau. \end{aligned} \tag{36}$$

Using the mean value theorem yields

$$|u(t_{n+1}) - u_{n+1}| \le |\gamma'(c_1, u(c_1))| \int_0^{t_{n+1}} (\tau - t_j)(t_{n+1}-\tau)^{\alpha-1}d\tau. \tag{37}$$

We calculate the above integral as follows:

$$\int_0^{t_{n+1}} (\tau - t_j)(t_{n+1} - \tau)^{\alpha-1} d\tau = \frac{(\Delta t)^{\alpha+1}}{\Gamma(\alpha+2)}((n+1)^\alpha(n+1-j(\alpha+1))). \tag{38}$$

Taking the limit of $|u(t_{n+1}) - u_{n+1}|$, we obtain

$$\lim_{\Delta t\to 0} |u(t_{n+1}) - u_{n+1}| = 0. \tag{39}$$

4.1 Error analysis with power-law kernel

In this section, we present error analysis for the considered numerical scheme. For error analysis, we consider a nonlinear fractional differential equation which is stated by

$$\begin{cases} {}_0^C D_t^\alpha u(t) = \gamma(t, u(t)), \\ u(0) = u_0. \end{cases} \tag{40}$$

To calculate the error, first, we suppose that the function $\gamma(t, u(t)) \in C^2([a, b])$. Denote the partition points as $t_j = a + jh$, $t_{j+1} = a + (j+1)h$, and $c_j = a + \frac{2j+1}{2^k}h$ the midpoints such that $j = 0,1,2,\ldots, n-1$. Before proceeding with Caputo case, we first start with classical case

$$\begin{aligned} u(t_{n+1}) - u(t_n) &= \int_{t_n}^{t_{n+1}} \gamma(\tau, u(\tau)) d\tau \\ &= \int_{t_n}^{t_{n+1}} \gamma(\tau, u(\tau)) d\tau \\ &= \int_{t_n}^{t_{n+1}} \left\{ \begin{matrix} P_n(\tau) \\ + \frac{(\tau - t_n)^2}{2!} \frac{\partial^2}{\partial\tau^2} \gamma(\tau, u(\tau)) \end{matrix} \right\} d\tau \\ &= \int_{t_n}^{t_{n+1}} \left\{ P_n(\tau) + \frac{(\tau - t_n)^2}{2!} \frac{\partial^2}{\partial\tau^2} \gamma(\tau, u(\tau)) \right\} d\tau \\ &\quad + E_n^\alpha, \end{aligned} \tag{41}$$

where

$$E_n^\alpha = \int_{t_n}^{t_{n+1}} \frac{(\tau - t_n)^2}{2!} \frac{\partial^2}{\partial\tau^2} \gamma(\tau, u(\tau)) d\tau. \tag{42}$$

Then, we write

$$\begin{aligned} E_n^\alpha &= \frac{1}{2} \sum_{j=1}^n \int_{t_j}^{t_{j+1}} (\tau - c_j)^2 \gamma''(c_j, u(c_j)) d\tau \\ &= \frac{1}{2} \sum_{j=1}^n \gamma''(c_j, u(c_j)) \sum_{m=0}^{2^k-1} \int_{t_{j+\frac{mh}{2^k}}}^{t_{j+\frac{(m+1)h}{2^k}}} (\tau - c_j)^2 d\tau. \end{aligned} \tag{43}$$

We evaluate

$$\sum_{m=0}^{2^k-1} \int_{t_{j+\frac{mh}{2^k}}}^{t_{j+\frac{(m+1)h}{2^k}}} (\tau - c_j)^2 d\tau = \sum_{m=0}^{2^k-1} \frac{h^3}{3 \times 2^{3k}}.$$

Then, for classical case, we obtain error as follows:

$$E_n^\alpha = \frac{1}{2} \sum_{j=1}^n \gamma''(\xi_j, u(\xi_j)) \frac{2^k h^3}{3 \times 2^{3k}}. \tag{44}$$

Considering $h = \frac{b-a}{n}$, then we have the following:

$$\begin{aligned} E_n^\alpha &= \frac{1}{6} \sum_{j=1}^n \gamma''(\xi_j, u(\xi_j)) \frac{h^3}{2^{2k}} \\ &\simeq \gamma''(\xi, u(\xi)) \frac{(b-a)h^2}{6 \times 2^{2k}}, \end{aligned} \tag{45}$$

where $\xi \in (a, b)$. When $k = 1$ which means we use midpoint, then we recover error for midpoint

$$E_n^\alpha \simeq \frac{(b-a)h^2}{24} \gamma''(\xi_j, u(\xi_j)). \tag{46}$$

Now, we continue with the Caputo case

$$\begin{aligned} u(t_{n+1}) - u(0) &= \frac{1}{\Gamma(\alpha)} \int_0^{t_{n+1}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} d\tau \\ &= \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} \gamma(\tau, u(\tau))(t_{n+1} - \tau)^{\alpha-1} d\tau \\ &= \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} \left\{ \begin{matrix} P_n(\tau) \\ + \frac{(\tau - t_m)^2}{2!} \frac{\partial^2}{\partial\tau^2} [\gamma(\tau, u(\tau))]_{\tau=\gamma_m} \end{matrix} \right\} \\ &\quad \times (t_{n+1} - \tau)^{\alpha-1} d\tau \\ &= \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^n \left[\begin{matrix} \sum_{m=0}^{2^{n-1}-1} \gamma\left(t_j + \frac{(2m+1)h}{2^n}, u\left(t_j + \frac{(2m+1)h}{2^n}\right)\right) \\ \times \left(\begin{matrix} \left(n - j + \left(1 - \frac{m}{2^{n-1}}\right)\right)^\alpha \\ - \left(n - j + \left(1 - \frac{m+1}{2^{n-1}}\right)\right)^\alpha \end{matrix} \right) \end{matrix} \right] + E_n^\alpha, \end{aligned} \tag{47}$$

where

$$E_n^\alpha = \frac{1}{\Gamma(\alpha)} \sum_{j=0}^n \int_{t_j}^{t_{j+1}} \frac{(\tau - c_j)^2}{2} \frac{\partial^2}{\partial\tau^2} [\gamma(\tau, u(\tau))]_{\tau=\gamma_m} \times (t_{n+1} - \tau)^{\alpha-1} d\tau. \tag{48}$$

We evaluate

$$
\begin{aligned}
E_n^\alpha &= \frac{1}{2\Gamma(\alpha)}\sum_{j=0}^{n}\gamma''(c_j,u(c_j))\int_{t_j}^{t_{j+1}}(\tau-c_j)^2(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau \\
&= \frac{1}{2\Gamma(\alpha)}\sum_{j=0}^{n}\gamma'^{'}(c_j,u(c_j)) \\
&\quad\times\sum_{m=0}^{2^k-1}\int_{t_{j+\frac{mh}{2^k}}}^{t_{j+\frac{(m+1)h}{2^k}}}(\tau-c_j)^2(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau.
\end{aligned}
\tag{49}
$$

If we take norm on both sides of the above equation, we write the following inequality:

$$
\begin{aligned}
&|E_n^\alpha| \\
&\le \frac{(\Delta t)^{\alpha+2}}{2\Gamma(\alpha+3)}\sup_{\tau\in[0,t_{n+1}]}\left|\frac{\partial^2}{\partial\tau^2}\gamma(\tau,u(\tau))\right| \\
&\sum_{m=0}^{2^k-1}\int_{t_{j+\frac{mh}{2^k}}}^{t_{j+\frac{(m+1)h}{2^k}}}(\tau-c_j)^2(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau,
\end{aligned}
\tag{50}
$$

where

$$
\begin{aligned}
&\sum_{m=0}^{2^k-1}\int_{t_{j+\frac{mh}{2^k}}}^{t_{j+\frac{(m+1)h}{2^k}}}(\tau-c_j)^2(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau \\
&= \left[\begin{array}{c}(n+1)^\alpha(6n^3+18n^2+18n+6) \\ -n^\alpha\begin{pmatrix}6n^3+18n^2+3\alpha^2+18n+9\alpha \\ +3n\alpha^2+6n^2\alpha+15n\alpha+6\end{pmatrix}\end{array}\right] \\
&\quad+2^{-k}\left[\begin{array}{c}n^\alpha\begin{pmatrix}6n^3\alpha+12+12n^3+6\alpha^2+18\alpha+36n^2 \\ +36n+6n^2\alpha^2+30n^2\alpha+12n\alpha^2+42n\alpha\end{pmatrix} \\ -(n+1)^\alpha\begin{pmatrix}6n^3\alpha+12+12n^3+6\alpha \\ +36n^2+36n+18n^2\alpha+18n\alpha\end{pmatrix}\end{array}\right] \\
&\quad+4^{-k}\left[((n+1)^\alpha-n^\alpha)\begin{pmatrix}(4n^3+12n^2+11n+3) \\ \times(\alpha^2+3\alpha+2)\end{pmatrix}\right].
\end{aligned}
\tag{51}
$$

Thus, the error can be obtained as

$$
|E_n^\alpha| \le \frac{(b-a)h^{\alpha+1}}{2\Gamma(\alpha+3)}\gamma''(\xi_j,u(\xi_j))(A+2^{-k}B+4^{-k}C),
\tag{52}
$$

where

$$
\begin{aligned}
A &= \left[\begin{array}{c}(n+1)^\alpha(6n^3+18n^2+18n+6) \\ -n^\alpha\begin{pmatrix}6n^3+18n^2+3\alpha^2+18n+9\alpha \\ +3n\alpha^2+6n^2\alpha+15n\alpha+6\end{pmatrix}\end{array}\right], \\
B &= \left[\begin{array}{c}n^\alpha\begin{pmatrix}6n^3(\alpha+2)+n^2(6\alpha^2+30\alpha+36) \\ +n(36+12\alpha^2+42\alpha) \\ +12+6\alpha^2+18\alpha\end{pmatrix} \\ -6(n+1)^\alpha(n+1)^3(\alpha+2)\end{array}\right], \\
C &= \left[((n+1)^\alpha-n^\alpha)\begin{pmatrix}(4n^3+12n^2+11n+3) \\ \times(\alpha^2+3\alpha+2)\end{pmatrix}\right].
\end{aligned}
\tag{53}
$$

5 Application of the proposed scheme to the Fitzhugh–Nagumo model

Simplifying the understanding of excitable media dynamics, the Fitzhugh–Nagumo equation is utilized to depict the behavior of action potentials in neurons and certain chemical reactions. They were proposed independently by Fitzhugh in 1961 [15] and Nagumo et al. in 1962 [16]. The solution of the Fitzhugh–Nagumo equation is of significant importance, as it allows for the acquisition of insights into the fundamental mechanisms underlying excitability in these cells. Therefore, in this section, we aim to solve this equation numerically. The associated model consists of two ordinary differential equations:

(1) The first equation represents the membrane potential V of the excitable cell:

$$
V'(t) = V - \frac{V^3}{3} - W + I.
\tag{54}
$$

(2) The second equation represents the recovery variable W, which models the inactivation of ion channels:

$$
W'(t) = \varepsilon(V + a - bW).
\tag{55}
$$

Here, the constant I represents an external stimulus and a, b, and ε are parameters. These equations capture the basic behavior of excitable systems, including the generation of action potentials and the refractory period. To achieve our aim, we first consider the Fitzhugh–Nagumo equation in the case of the Atangana–Baleanu fractional derivative

$$
\begin{aligned}
{}_0^{AB}D_t^\alpha V(t) &= \varepsilon\left(V - \frac{V^3}{3} - W + I\right), \\
{}_0^{AB}D_t^\alpha W(t) &= V + a - bW,
\end{aligned}
\tag{56}
$$

where the initial conditions are $V(0) = -0.5$, $W(0) = -0.1$ and the parameters are chosen as $a = 0.001$, $b = 0.002$, $\varepsilon = 9$, and $I = 0.32$. Applying the associated integral on the above equation and evaluation at the point $t = t_{n+1}$, we have the following:

$$
\begin{aligned}
V(t_{n+1}) &= V(0) + (1-\alpha)\gamma_1(t_{n+1}, V(t_{n+1}), W(t_{n+1})) \\
&\quad+\frac{\alpha}{\Gamma(\alpha)}\int_0^{t_{n+1}}\gamma_1(\tau, V(\tau), W(\tau))(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau \\
W(t_{n+1}) &= W(0) + (1-\alpha)\gamma_2(t_{n+1}, V(t_{n+1}), W(t_{n+1})) \\
&\quad+\frac{\alpha}{\Gamma(\alpha)}\int_0^{t_{n+1}}\gamma_2(\tau, V(\tau), W(\tau))(t_{n+1}-\tau)^{\alpha-1}\mathrm{d}\tau,
\end{aligned}
\tag{57}
$$

where

$$\begin{aligned} \gamma_1(t,V,W) &= V - \frac{V^3}{3} - W + I, \\ \gamma_2(t,V,W) &= \varepsilon(V + a - bW). \end{aligned} \tag{58}$$

Employing the suggested method yields

$$\begin{aligned} V_{n+1} = V(0) &+ (1-\alpha)\gamma_1(t_{n+1}, \widetilde{V}_{n+1}, \widetilde{W}_{n+1}) \\ &+ \frac{h^\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\left[\sum_{m=0}^{2^{k-1}-1}\gamma_1\left(t_j + \frac{(2m+1)h}{2^k}, V\left(t_j + \frac{(2m+1)h}{2^k}\right), W\left(t_j + \frac{(2m+1)h}{2^k}\right)\right) \times \left(\left(n - j + \left(1 - \frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n - j + \left(1 - \frac{m+1}{2^{k-1}}\right)\right)^\alpha\right)\right] \end{aligned} \tag{59}$$

$$\begin{aligned} W_{n+1} = W(0) &+ (1-\alpha)\gamma_2(t_{n+1}, \widetilde{V}_{n+1}, \widetilde{W}_{n+1}) \\ &+ \frac{h^\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\left[\sum_{m=0}^{2^{k-1}-1}\gamma_2\left(t_j + \frac{(2m+1)h}{2^k}, V\left(t_j + \frac{(2m+1)h}{2^k}\right), W\left(t_j + \frac{(2m+1)h}{2^k}\right)\right) \times \left(\left(n - j + \left(1 - \frac{m}{2^{k-1}}\right)\right)^\alpha - \left(n - j + \left(1 - \frac{m+1}{2^{k-1}}\right)\right)^\alpha\right)\right]. \end{aligned}$$

The predictor components are calculated as

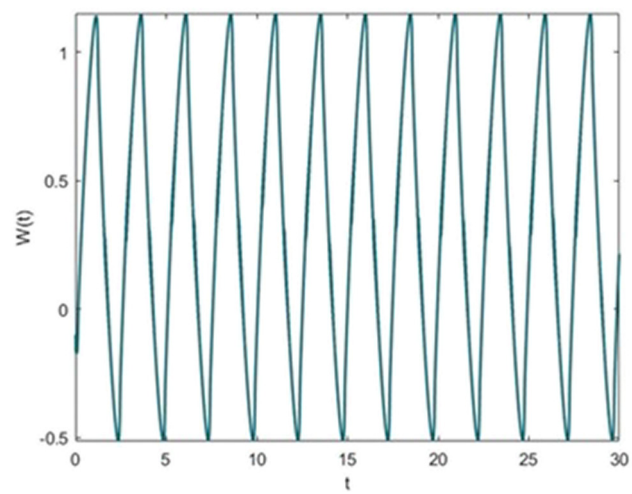

Figure 4: Numerical simulation of the Fitzhugh–Nagumo equation with the Atangana–Baleanu fractional derivative for $\alpha = 0.9$, $k = 4$.

$$\begin{aligned} \widetilde{V}_{n+1} &= V_0 + (1-\alpha)\gamma_1(t_n, V_n, W_n) + \frac{h^\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\gamma_1(t_j, V_j, W_j) \times ((n-j+1)^\alpha - (n-j)^\alpha), \\ \widetilde{W}_{n+1} &= W_0 + (1-\alpha)\gamma_2(t_n, V_n, W_n) + \frac{h^\alpha}{\Gamma(\alpha)}\sum_{j=0}^{n}\gamma_2(t_j, V_j, W_j) \times ((n-j+1)^\alpha - (n-j)^\alpha). \end{aligned} \tag{60}$$

In Figures 3–5, we present the numerical simulation of the Fitzhugh–Nagumo equation with the Atangana–Baleanu fractional derivative.

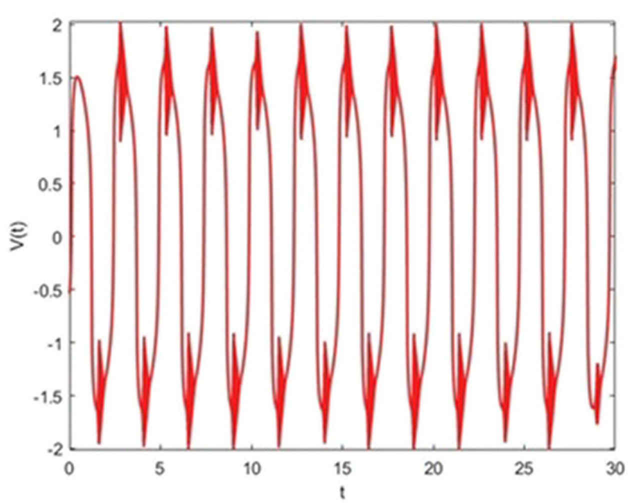

Figure 3: Numerical simulation of the Fitzhugh–Nagumo equation with the Atangana–Baleanu fractional derivative for $\alpha = 0.9$, $k = 4$.

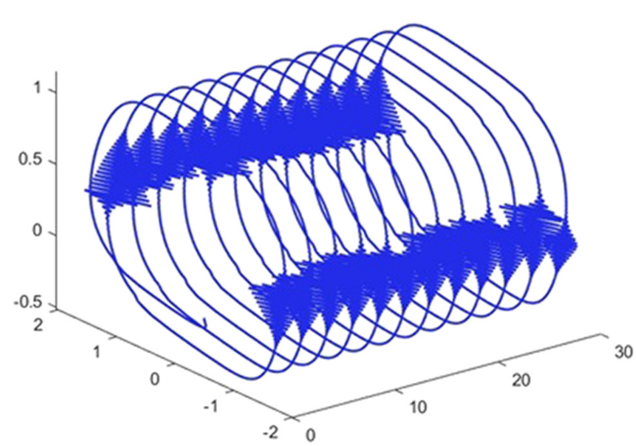

Figure 5: Numerical simulation of the Fitzhugh–Nagumo equation with the Atangana–Baleanu fractional derivative for $\alpha = 0.9$, $k = 4$.

6 Conclusion

This study presents the derivation and analysis of the successive midpoint method for solving fractional differential equations involving the Atangana–Baleanu and Caputo derivatives. While the method has previously been applied to classical and Caputo–Fabrizio fractional models, this work extends its applicability to a broader class of fractional operators. A comprehensive error analysis was conducted, along with investigations into the method's consistency and stability for both classical and Caputo-type equations.

Furthermore, numerical experiments were performed to evaluate the effectiveness of the successive midpoint method in comparison to the well-known midpoint method and L1 method. The results demonstrate that the successive midpoint method consistently outperforms the classical midpoint scheme and L1 method in terms of accuracy. Notably, its application to equations involving the Atangana–Baleanu derivative proved particularly advantageous, owing to the integration of a dedicated predictor mechanism tailored to the non-local and non-singular nature of this derivative. Overall, this work contributes to the growing body of the literature on numerical methods for fractional differential equations by offering a robust, accurate, and practically implementable technique. Given the complex and memory-dependent structure of fractional models, the successive midpoint method stands out as a promising tool for researchers and practitioners working with advanced fractional dynamics.

The successive midpoint method offers several advantages, including improved accuracy, consistency, and stability over the classical midpoint method. Its performance is especially notable for equations involving the Atangana–Baleanu derivative, as it incorporates a tailored predictor component that effectively handles the non-local and non-singular nature of this operator. This makes it a robust tool for modeling complex fractional systems with memory effects. However, the method also has some limitations. It is computationally more demanding due to its predictor–corrector structure and may require careful tuning of parameters, such as step size, to maintain efficiency.

Acknowledgments: The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at the Qassim University for financial support (QUAPC2025).

Funding information: The financial support by the Deanship of Graduate Studies and Scientific Research at the Qassim University (QUAPC2025).

Author contributions: Maroua Amel Boubekeur: conceptualization, investigation; visualization; roles/writing – original draft; writing – review, and editing. Salah Boulaaras: conceptualization, investigation, methodology, software, supervision, visualization, writing – review, and editing. Seda İğret Araz: conceptualization, investigation, methodology, software, supervision, visualization, writing – review, and editing. All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interests.

Data availability statement: All data generated or analysed during this study are included in this published article.

References

[1] Endre S, David M. An introduction to numerical analysis. Cambridge, United Kingdom: Cambridge University Press, ISBN 0-521-00794-1, 2003.
[2] Ralston A. Runge–Kutta methods with minimum error bounds. Math Comput. 1962;92:339.
[3] Argyros IK, Chen D. The midpoint method for solving nonlinear operator equations in Banach space. Appl Math Lett. 1992;5:7–9.
[4] Toufik M, Atangana A. New numerical approximation of fractional derivative with non-local and non-singular kernel: application to chaotic models. Europ Phys J Plus. 2017;132:1–16.
[5] Atangana A, Igret Araz S. New numerical scheme with newton polynomial: theory, methods, and applications. London, England: Elsevier, Academic Press; 2021.
[6] Arık IA, Sarı HK, Igret Araz S. Numerical simulation of Covid-19 model with integer and non-integer order: The effect of environment and social distancing. Results Phys. 2023;51:106725.
[7] Iğret Araz S, Çetin MA. Fractal-fractional modeling of the Covid-19 spread with deterministic and stochastic approaches. Int J Appl Comput Math. 2025;11:4.
[8] Yazdani AB, Kiasari MM, Jafari H. A novel approach for solving time fractional nonlinear Fisher's equation by using Chebyshev spectral collocation method. Progress Fract Differ Appl. 2021;7(2):97–102.
[9] Ali KK, Karakoç SBG, Rezazadeh H. Optical soliton solutions of the fractional perturbed nonlinear Schrodinger equation, TWMS J Appl Eng Math. 2020;10(4):930–9.
[10] Karakoç SBG, Bhowmik SK, Omrani K, Yıldırım Sucu D. A numerical treatment for the Gilson-Pickering equation using collocation method with error estimation. Sigma J Eng Nat Sci. 2023;41(4):875–84.
[11] Atangana A, Igret Araz S. A successive midpoint method for nonlinear differential equations with classical and Caputo-Fabrizio derivatives. AIMS Math. 2023;8(11):27309–27.
[12] Podlubny I. Fractional Differential Equations, vol. 198 of Mathematics in Science and Engineering. San Diego: Academic Press; 1999. ISBN: 9780125588409.

[13] Caputo M, Linear model of dissipation whoseQ is almost frequency independent. II. Geophys J Int. 1967;13(5):529–39.

[14] Atangana A, Baleanu D New fractional derivatives with non-local and non-singular kernel. Theory Appl Heat Transfer Model Thermal Sci. 2016;20(2):763–9.

[15] FitzHugh R. Impulses and physiological states in theoretical models of nerve membrane. Biophys J. 1961;1(6):445–66.

[16] Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. Proc IRE. 1962;50(10):2061–70.

[17] Atangana A, Igret Araz S. Theory and methods of piecewise defined fractional operators. Elsevier. 2025.

[18] Arnold DN. Stability, consistency, and convergence of numerical discretizations, encyclopedia of applied and computational mathematics. Berlin, Heidelberg: Springer; 2015. p. 1358–64.

[19] Eslahchi MR, Esmaili S. The convergence and stability analysis of a numerical method for solving a mathematical model of language competition. Appl Numer Math. 2020;151:119–40.

[20] Li M, Shi D, Pei L. Convergence and superconvergence analysis of finite element methods for the time fractional diffusion equation. Appl Numer Math. 2020;151:141–60.

[21] Costa VAF. On the convergence of the iterative methods. Numer Heat Transfer Part B Fund. 2022;81:1–6, 1–13.

[22] Wang J, Yang JZ, Zhang J. Stability and convergence analysis of high-order numerical schemes with DtN-type absorbing boundary conditions for nonlocal wave equations. IMA J Numer Anal. 2024;44(1):604–32.

[23] Franyelit SC, Luis RR. Convergence and stability criteria for numerical solutions of partial differential equations in science and engineering. Univ Ciencia Y Tech. 2024;28(122):7–16.

[24] Nasresfahani F, Eslahchi MR. Numerical convergence and stability analysis for a nonlinear mathematical model of prostate cancer. Numer Methods Partial Differ Equ. 2023;39(4):3064–88.