e-ISSN: 2149-4584

Erzincan University Journal of Science and Technology

EJSAT 2025, 18(3), 726–734

Research Article

Physics Informed Neural Network Method for the Numerical Solution of Fractional Diffusion Equations

Mehmet Fatih UÇAR[1*], Burcu Ece ALP[2]

[1*]Department of Mathematics and Computer Science, Istanbul Kultur University, Türkiye

[2]Turkcell Head Office Software Unit Data Analyst Istanbul, Türkiye

Received: 26/10/2024, **Revised:** 13/01/2025, **Accepted:** 23/01/2025, **Published:** 31/12/2025

Abstract

Artificial neural networks are gaining popularity for developing continuous solution functions to address various types of differential equations. In this research, we introduce a physics-informed neural network (PINN) approach for solving fractional diffusion equations with variable coefficients over a finite domain. The PINN framework approximates solutions to fractional PDEs by optimizing a physical loss function that incorporates terms for the residual, boundary conditions, and initial conditions. The fractional PDE is discretized using the Grunwald-Letnikov scheme, and the resulting semi-discrete equation is employed to define the residual term for the PINN. Numerical tests confirm that the proposed PINN method achieves precise solutions within the defined computational space-time domain.

Keywords: 35R11 Fractional partial differential equations, 65N22 Numerical solution of boundary value problems, 68T07 Artificial neural networks and deep learning.

Kesirli Difüzyon Denklemlerinin PINN Metodu ile Sayısal Çözümleri

Öz

Yapay sinir ağları, çeşitli diferansiyel denklemlerin çözümünde sürekli çözüm fonksiyonları oluşturmak için giderek daha fazla kullanılmaktadır. Bu çalışmada, sonlu bir alan üzerinde değişken katsayılı kesirli difüzyon denklemlerini fizik bilgili sinir ağı (PINN) yöntemi ile sayısal çözümlerini ele alacağız. PINN, rezidü, sınır koşulu ve başlangıç koşulundan oluşan fiziksel hata fonksiyonunun değerini en aza indirmek için eğitilerek kesirli PDE'ye yaklaşık çözümler üretir. Kesirli PDE, Grünwald-Letnikov formülü ile ayrıklaştırılır ve elde edilen yarı ayrık denklem, PINN'nin rezidü fonksiyonunu oluşturmak için kullanılır. Ele aldığımız denklemler, mevcut PINN yönteminin dikkate alınan hesaplama uzay-zaman alanı üzerinde doğru çözümler sağladığını göstermektedir.

Anahtar Kelimeler: 35R11 Kesirli türevli kısmi diferansiyel denklemler, 65N22 Sınır değer problemlerinin sayısal çözümleri, 68T07 Yapay sinir ağları ve derin öğrenme.

Corresponding Author: f.ucar@iku.edu.tr

Cite this Article: Erzincan University Journal of Science and Technology 2025, 18(3), 726–734.
https://doi.org/10.18185/erzifbed.1589798

1. Introduction

Neural networks have proven to be highly effective in addressing problems that exhibit nonlinearity. Their ability to tackle nonlinear challenges improves as the number of layers and neurons increases. One application of neural networks involves predicting bankruptcy. Recently, various methods leveraging neural networks have emerged for solving complex nonlinear differential equations. For instance, Mall and Chakraverty [1] proposed an artificial neural network (ANN) approach. Sabir et al. utilized a stochastic numerical algorithm rooted in neural networks to solve boundary value problems [2]. Raissi et al. introduced the Physics-Informed Neural Network (PINN) methodology, which incorporates physics-based constraints [3]. This innovative technique has been applied to both ordinary and partial differential equations. Additionally, Dwivedi et al. devised a distributed learning machine approach to address partial differential equations [4-7].

Fractional calculus has emerged as a significant and dynamic area of research in applied mathematics [8–10]. This field encompasses a broad range of fractional-order operators, including both derivatives and integrals. Recently, a novel definition of the fractional derivative based on a nonsingular kernel was introduced in [11, 12]. Additionally, [13] proposed a new fractional derivative framework utilizing the Mittag-Leffler kernel. Such operators are widely applied to address various challenges in science and engineering. Rostami et al. developed a method to solve high-order fractional differential equations using artificial neural networks (ANN) [14]. Heuristic algorithms have been employed to optimize these complex systems [15]. Alkan et al. investigated different forms of fractional partial differential equations (PDEs) [16–19], while Bektas et al. presented a hybrid approach for solving the fractional NWS equation [20]. Pakdaman et al. combined an ANN with an optimization technique to tackle fractional differential equations [21].

Moreover, the application of ANN to approximate solutions for fractional-order partial differential equations has been extensively studied. A Physics-Informed Neural Network (PINN) approach was extended to fractional problems, introducing an innovative neural network framework [22]. The PINN method has been particularly effective for solving fractional diffusion equations, addressing challenges like prolonged training times and slow convergence that often hinder traditional neural network models. By training the network to minimize a physical loss function composed of residual, boundary, and initial condition terms, the PINN method offers a robust tool for approximating solutions to fractional PDEs.

In this research, the fractional derivative term is approximated using the first-order Grünwald-Letnikov formula, and the resulting semi-discrete equation is utilized to formulate the residual function for the PINN. Numerical results demonstrate that the proposed PINN approach delivers precise solutions within the specified space-time computational domain.

2. Material and Methods

2.1. Fractional Diffusion Equation

We consider the following diffusion equation of order $1 < \beta < 2$

$$\frac{\partial U(x,t)}{\partial t} - d_+(x)\frac{\partial^\beta U(x,t)}{\partial_+ x^\beta} - d_-(x)\frac{\partial^\beta U(x,t)}{\partial_- x^\beta} = f(x,t), \tag{1}$$

$$x_L < x < x_R \qquad 0 < t \leq T,$$
$$U(x_L, t) = 0, \qquad U(x_R, t) = 0, \qquad 0 \leq t \leq T$$
$$U(x, 0) = U_0(x), \qquad x_R \leq x \leq x_L$$

and the Dirichlet boundary condition as

$$U(x_L, t) = 0 \quad \text{and} \quad U(x_R, t) = \xi(t)$$

where the left-sided (-) and the right-sided (+) fractional derivatives $\frac{\partial^\beta U(x,t)}{\partial_- x^\beta}$ and $\frac{\partial^\beta U(x,t)}{\partial_+ x^\beta}$ can be written in the Grünvald-Letkinov form:

$$\frac{\partial^\beta U(x_i, t^m)}{\partial_- x^\beta} = \frac{1}{(\Delta x)^\beta} \sum_{k=0}^{i+1} h_k^{(\beta)} U_{i-k+1}^m + O(h), \tag{2}$$

$$\frac{\partial^\beta U(x_i, t^m)}{\partial_+ x^\beta} = \frac{1}{(\Delta x)^\beta} \sum_{k=0}^{N-i+1} h_k^{(\beta)} U_{i+k-1}^m + O(h). \tag{3}$$

Here, $h_k^{(\beta)} = (-1)^k \begin{pmatrix} \beta \\ k \end{pmatrix} \frac{\Gamma(k-\beta)}{\Gamma(-\beta)\Gamma(k+1)}$ where $\begin{pmatrix} \beta \\ k \end{pmatrix}$ is the fractional binomial coefficient. Furthermore, the coefficients $h_k^{(\beta)}$ can be evaluated in recurrence relation

$$h_0^{(\beta)} = 1, \qquad h_k^{(\beta)} = (1 - \frac{\beta+1}{k}) h_{k-1}^{(\beta)} \qquad k \geq 1. \tag{4}$$

The case $1 < \beta < 2$, it is beneficial in applications.

2.2. Physics Informed Neural Network(PINN)

(PINNs) are a deep learning framework designed to solve partial differential equations (PDEs) [3]. This approach enables computers to efficiently and accurately compute partial derivatives of functions. The time variable, t, is treated as a specific component of x, while Θ represents the spatial domain. The initial condition can be viewed as a special case of a Dirichlet boundary condition applied to the spatio-temporal domain, making the process relatively straightforward. By training a neural network to minimize a loss function, which incorporates terms for the mismatch of initial and boundary conditions along the space-time boundary as well as the PDE residual at selected interior points, PINNs approximate solutions to PDEs. Furthermore, they allow for direct differentiation of functions.

Let us begin by analyzing a problem expressed as follows:

$$\mathcal{L}U(x,t) = f_{BB}(x,t), \qquad (x,t) \in \Theta \times (0,T], \tag{5}$$
$$U(x,0) = g(x), \qquad x \in \Theta, \tag{6}$$
$$U(x,t) = 0, \qquad x \in \partial\Theta, \tag{7}$$

where the function $g(.)$ represents the initial condition. The proposed approximate solution is defined as

$$\tilde{U}(x,t) = t\rho(x) U_{NN}(x,t;\mu) + g(x), \tag{8}$$

which inherently satisfies the initial and boundary conditions. Here, $L(.)$ denotes either a linear or nonlinear operator.

By defining $z = [x, t]^T$, the PINN method assumes the following approximate function:

$$U_\theta(z) := W^L \sigma^L (W^{L-1} \sigma^{L-1} (...\sigma^1 (W^0 z + b^0)) + b^{L-1}) + b^L, \tag{9}$$

where θ represents the set of weight matrices (W) and bias vectors (b). The weight matrices are specified as $W^0 \in R^{m\times 2}$, $W^i \in R^{m\times m}$ $(i = 1, 2, ..., L-1)$ and $W^L \in R^{1\times m}$ where m is the number of neurons and L denotes the total number of layers. The bias vectors are defined as $b^i \in R^{m\times 1}(i = 1, 2, ..., L-1)$ and $b^L \in R$. The activation function $\sigma(.)$ transforms the network into a nonlinear model. The expression in Eq. (9) serves as an approximate function for a multilayer feedforward neural network. This function depends on the parameter set $\theta = W^0, b^0, W^1, b^1, ..., W^L, b^l$, which must be optimized to solve the fractional diffusion equation. The continuous residual function used in this optimization process is expressed as:

$$R_\theta(x, t) := \theta_t u_\theta(x, t) - L[u_\theta](x, t) + q(x, t), \tag{10}$$

which is derived by PINN from Eqs. (2) and (3).

The PINN method minimizes the residual function over the collocation points defined in the region of the fractional differential equation. Simultaneously, the network adjusts the θ parameters to satisfy both boundary and initial conditions. Suppose the collocation points are given as:

$$X_i^r = \{(x_i^r, t_i^r)\}_{i=1}^{N_r} \subset [x_L, x_R]\text{x}[0, T] \tag{11}$$

that is generetad by PINN from Eqs. (2) and (3).

The PINN methodology seeks to minimize the residual function at collocation points within the domain of the fractional differential equation, while simultaneously optimizing the parameter θ to satisfy both boundary and initial conditions. Let the collocation points be defined as follows:

In the standard PINN approach, collocation points are typically chosen randomly. However, in this case, they are taken as uniformly spaced points. This is due to the derivative approximations in equations (2) and (3), which require x_i points with a step size h.

The PINN approach minimizes the following loss functional to solve Eq. (1):

$$\phi_\theta := \phi_\theta^r(X^r) + \phi_\theta^b(X_b^r) + \phi_\theta^s(X_s^r). \tag{12}$$

Here, the functional comprises three distinct loss components corresponding to the residual, initial condition, and boundary conditions. The Nadam optimization algorithm is employed to minimize these loss terms. Specifically, the residual loss function is expressed as:

$$\phi_\theta^r(X^r) = \frac{1}{N_r} \sum_{i=1}^{N_r} |r_\theta(x_i^r, t_i^r)|^2. \tag{13}$$

Meanwhile, the loss functions for initial and boundary conditions are defined as:

$$\phi_\theta^b(X_b^r) = \frac{1}{N_b} \sum_{i=1}^{N_b} |u_\theta(x_i^b, 0) - g(x_i^b)|^2. \tag{14}$$

$$\phi_\theta^s(X_s^r) = \frac{1}{N_s}\sum_{i=1}^{N_s} |u_\theta(x_i^s, t_i^s) - u_b(x_i^s, t_i^s)|^2. \tag{15}$$

The boundary condition function $u_b(x_i^s, t_i^s)$ is given by:

$$U_b(x_i^s, t_i^s) = \begin{cases} 0, & \text{if } x_i^s = x_L, \\ 1, & \text{if } x_i^s = x_R. \end{cases} \tag{16}$$

Thus, solving the optimization problem:

$$\Theta^* = \arg\min_{\phi} \theta(x^R), \tag{17}$$

yields the optimal parameters ϕ^*, which minimize the initial and boundary loss terms.
For this study, the implementation was carried out using the TensorFlow Python library, and the optimal parameters ϕ^* were obtained via the Nadam optimization algorithm.

3. Numerical Example

3.1. Example 1

Let us consider the fractional diffusion equation (1) with

$$d_+(x) = \Gamma(1.2)x^{1.8} \quad \text{and} \quad d_-(x) = \Gamma(1.2)(2-x)^{1.8}.$$

The spatial domain is $[x_R, x_L] = [0, 2]$ and the time interval is $[0, T] = [0, 1]$. The source term and the initial condition are given by

$$f(x,t) = -32e^{-t}[x^2 + (2-x)^2 + 0.125x^2(2-x)^2 - 2.5(x^3 + (2-x)^3) + \frac{25}{22}(x^4 + (2-x)^4)] \tag{18}$$

$$u_0(x) = 4x^2(2-x)^2 \tag{19}$$

respectively. The exact solution to the corresponding fractional diffusion Eq. (1) is given by [18]

$$u(x,t) = 4e^{-t}x^2(2-x)^2. \tag{20}$$

In the numerical experiment, we apply the PINN method with $N = 20$ and $L = 1$ to solve the given equation numerically and absolute errors are presented in Table 1 for different spatial mesh sizes and fixed time step. Also we compare the results with fast forward difference and forward difference results that is obtained in [23].

Table 1: Maximum absolute errors, $\Delta t = 1/128$

n	PINN	FD[23]	FFD[23]
2^6	8.59x10^{-4}	1.74x10^{-2}	1.55x10^{-2}
2^7	2.98x10^{-4}	8.35x10^{-3}	7.02x10^{-3}
2^8	7.25x10^{-5}	4.08x10^{-3}	3.18x10^{-3}

3.2. Example 2

Let us analyze the following other fractional diffusion equation from [24] with

$$d_+(x) = d_-(x) = \frac{\Gamma(2.2)x^{2.8}}{6}. \tag{21}$$

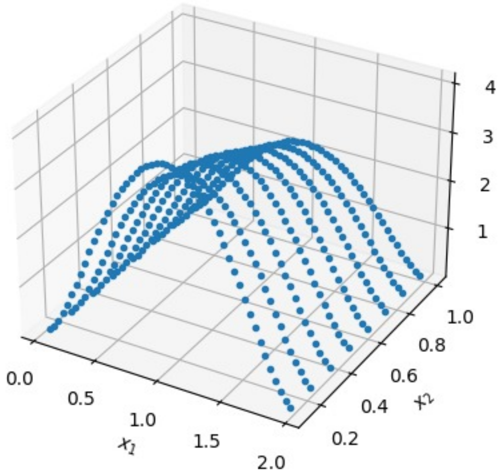

Figure 1: Graph of Ex.1 with $N = 2^7$ and $\Delta t = 1/128$.

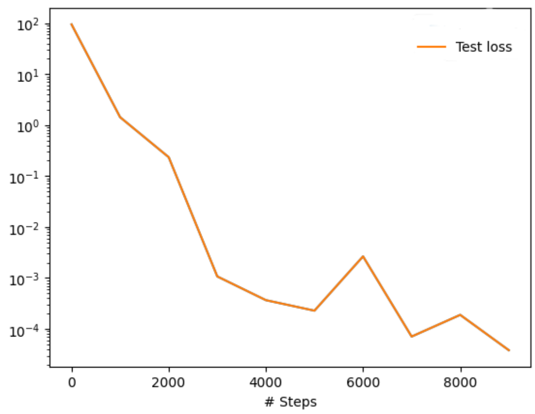

Figure 2: Behaviour of Error for Ex.1.

The spatial domain is $[x_R, x_L] = [0, 1]$ the time interval is $t > 0$. The source function is given by

$$f(x,t) = (1 + x)e^{-t}x^3,$$

and the initial condition is given by

$$u(x, 0) = x^3, \quad 0 < x < 1.$$

Dirichlet boundary conditions are

$$u(1,t) = 0 \quad \text{and} \quad u(1,t) = e^{-t}, \quad t > 0.$$

The exact solution to the corresponding fractional diffusion equation is given by $u(x,t) = e^{-t}x^3$.

In the numerical experiment, we reapply PINN method with N=5 and N=20 and L=1 to solve the given equation numerically and absolute errors are presented in Table 2 for different spatial mesh sizes and fixed time step. Also we compare the results with non polynomial spline method [24] and local polynomial regression (LPR) method [25].

Table 2: Maximum absolute errors, $\Delta t = 0.01$

n	PINN (N=5, L=1)	PINN (N=20, L=1)	Spline M.[24]	LPR M.[25]
11	0.03687	0.01125	0.1762	0.10446
21	0.01482	0.009588	0.1663	0.10518
61	0.008754	0.005224	0.1307	
121	0.001547	0.0009874	0.0859	

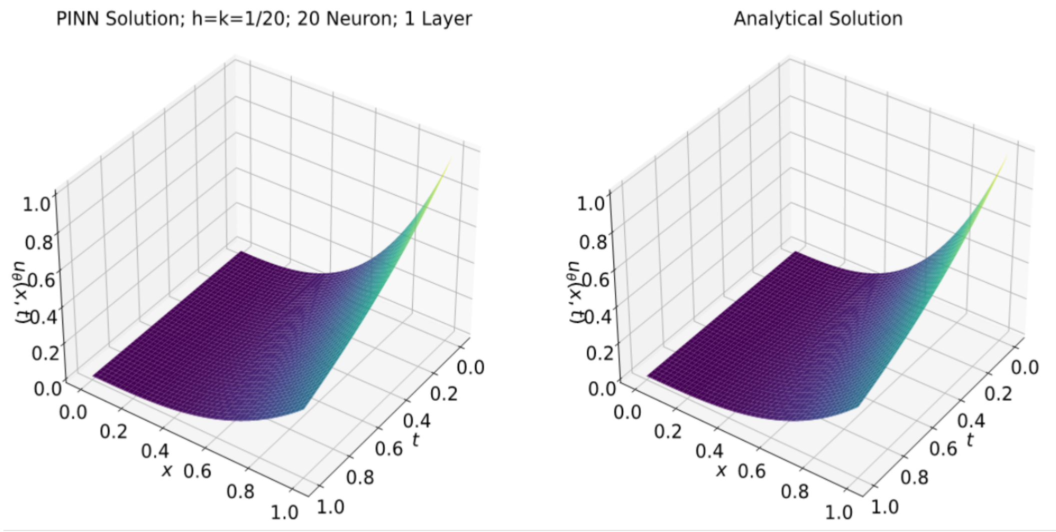

Figure 3: Solution graphs of Ex.2

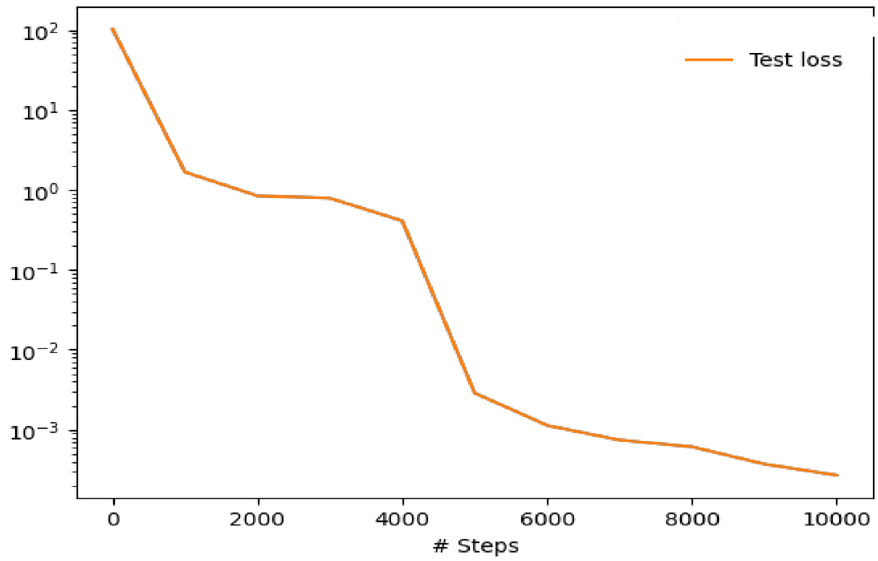

Figure 4: Behaviour of Error for Ex.2

4. Conclusion

This research presents approximate solutions and error analysis for the fractional diffusion equations, achieved through the feed-forward PINN model. Unlike traditional numerical methods, PINNs utilize automatic differentiation to manage differential operators. In contrast to numerical differentiation, automatic differentiation avoids differentiating the data itself. The neural networks designed to solve these problems were implemented using the Nadam optimizer and the Sigmoid activation function within the TensorFlow framework. Based on numerical experiments, it has been concluded that the PINN method offers a promising, convergent alternative. Future research will focus on applying the neural network approach to solve higher-order fractional partial differential equations (PDEs) numerically.

Competing interests:

The authors do not have any competing interests in this research.

Author contributions:

The authors contributed equally to all parts of the article.

Availability of data and materials:

Data sharing is not applicable to this article, as no datasets were generated or analyzed during the current study.

References

[1] Mall S., Chakraverty S.,(2013) Comparison of artificial neural network architecture in solving ordinary differential equations, Adv. Artif. Neural Syst., 2013 , p. 12

[2] Sabir Z., Baleanu D., Shoaib M., Raja M.A.Z., (2021) Design of stochastic numerical solver for the solution of singular three-point second-order boundary value problems, Neural Comput. Appl., 33 (7) , pp. 2427-2443.

[3] Raissi M., Perdikaris P., Karniadakis G.E., (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys., 378 , pp. 686-707.

[4] Dwivedi V., Parashar N., Srinivasan B., (2021) Distributed learning machines for solving forward and inverse problems in partial differential equations, Neurocomputing, 420 , pp. 299-316.

[5] Cai S., Wang Z., Wang S., Perdikaris P., Karniadakis G.E., (2021) Physics-informed neural networks for heat transfer problems, J. Heat Transfer, 143 (6) .

[6] Mortari D., (2017) The theory of connections: Connecting points, Mathematics, 5 (4) , p. 57.

[7] Mai T., Mortari D., (2022) Theory of functional connections applied to quadratic and nonlinear programming under equality constraints, J. Comput. Appl. Math., 406 .

[8] Li, X. (2012) Numerical solution of fractional differential equations using cubic B-spline wavelet collocation method, Commun Nonlinear Sci Numer Simulat 17 3934–3946.

[9] Diethelm K., Ford N.J., (2002) Analysis of fractional differential equations, J. Math. Anal. Appl., 265 (2) , pp. 229-248.

[10] Kilbas A.A., Srivastava H.M., Trujillo J.J., (2006) Theory and Applications of Fractional Differential Equations, Vol. 204 Elsevier

[11] Podlubny I., (1998) Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution and Some of their Applications Elsevier .

[12] Caputo M., Fabrizio M. A new definition of fractional derivative without singular kernel Prog. Fract. Differ. Appl., 1 (2) , pp. 73-85

[13] Atangana A., Baleanu D., (2015) New fractional derivatives with non-local and non-singular kernel, theory and application to heat transfer model, Therm. Sci., 20 (2) (2016), Article 763769.

[14] Jafarian A., Mokhtarpour M., Baleanu D., (2017) Artificial neural network approach for a class of fractional ordinary differential equation, Neural Comput. Appl., 28 (4) , pp. 765-773.

[15] Sivalingam S.M., Kumar P., Govindaraj V., (2023) The hybrid average subtraction and standard deviation based optimizer, Adv. Eng. Softw., 176 , Article 103387.

[16] Alkan, A. (2022). Improving homotopy analysis method with an optimal parameter for time-fractional Burgers equation. Karamanoğlu Mehmetbey Üniversitesi Mühendislik ve Doğa Bilimleri Dergisi, 4(2), 117-134.

[17] Alkan, A. (2022). Improving Homotopy Analysis Method with An Optimal Parameter for Time-Fractional Burgers Equation. Karamanoğlu Mehmetbey Üniversitesi Mühendislik Ve Doğa Bilimleri Dergisi, 4(2), 117-134. https://doi.org/10.55213/kmujens.1206517

[18] Alkan, A., & Anaç, H. (2024). The novel numerical solutions for time-fractional Fornberg-Whitham equation by using fractional natural transform decomposition method. AIMS Mathematics, 9(9), 25333-25359.

[19] Alkan, A., & Anaç, H. (2024). A new study on the Newell-Whitehead-Segel equation with Caputo-Fabrizio fractional derivative. AIMS Mathematics, 9(10), 27979-27997.

[20] Bektaş, U., & Anaç, H. (2024). A hybrid method to solve a fractional-order Newell–Whitehead–Segel equation. Boundary Value Problems, 2024(1), 38.

[21] Pakdaman M., Ahmadian A., Effati S., Salahshour S., Baleanu D., (2017) Solving differential equations of fractional order using an optimization technique based on training artificial neural network, Appl. Math. Comput., 293 , pp. 81-95.

[22] Pang G., Lu L., Karniadakis G.E., (2019) fPINNs: Fractional physics-informed neural networks, SIAM J. Sci. Comput., 41 (4) , pp. A2603-A2626.

[23] Wang, W., Wang K., Sircar, T. (2019) A direct $0(Nlog^2N)$ finite difference method for fractional diffusion equations, Journal of Computational Physics, 229 8095-8104.

[24] Caglar, S., Ucar, M.F., Caglar, N., Akkoyunlu, C., (2012) Non Polynomial Spline Method for Fractional Diffusion Equation, Journal of Computational Analysis and Applications, 1354 - 1361.

[25] Tadjeran, C., Meerschaert, M. M., (2007) A second-order accurate numerical method for the two-dimensional fractional diffusion equation, Journal of Computational Physics, 220 813–823.