The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

A Modified Fractional Physics-Informed Neural Networks for Solving Fractional Reaction-Diffusion Problems

Maryam Mohammadi* Mohadese Ramezani† Reza Mokhtari‡

Abstract

This paper introduces an innovative approach featuring a modified fractional Physics-Informed Neural Network (fPINN) that effectively tackles the challenges associated with fractional convection-diffusion problems. These problems often pose difficulties for traditional numerical methods, especially in high-dimensional spaces or complex geometries. Numerical experiments were conducted using a well-defined 2D benchmark example to demonstrate the effectiveness of the proposed framework. The results indicate that this framework significantly improves the performance of radial basis function neural networks, making them better suited for handling complex fractional models.

Keywords: fractional PINN, fractional convection-diffusion equations, weak singular solution

1 Introduction

Deep neural networks (DNNs) have made significant advancements in various fields, including computer vision, natural language processing, and game theory. Their remarkable capability to approximate complex functions has led to widespread application in tackling difficult problems in applied mathematics, particularly in solving partial differential equations (PDEs) and fractional PDEs.

Fractional PDEs are notable for their non-local behavior, making them effective models for processes that involve memory and hereditary effects. However, obtaining analytical solutions for fractional PDEs is often challenging due to the complexity of special functions, such as the Mittag-Leffler and Wright functions. Consequently, various numerical methods have been developed over the years to approximate these solutions such as finite difference, finite element, and spectral methods. Despite their usefulness, numerically solving fractional PDEs presents significant challenges. The high computational costs and memory requirements associated with the non-local behavior and singularities of fractional derivatives complicate the process. Traditional numerical methods, particularly for high-dimensional fractional PDEs, face substantial obstacles because they depend heavily on grid discretization, mesh generation, and the incorporation of physical laws, which increase computational costs and complexity.

Given these limitations, deep learning approaches, particularly physics-informed neural networks (PINNs), have emerged as promising alternatives. PINNs, developed by Raissi et al. [5], offer a framework for solving PDEs by directly incorporating the physical laws described by these equations into the learning process through the loss function. The key idea is to adjust the loss function to embed these physical laws within the neural network, reconciling inconsistencies in data, such as initial and boundary conditions or scattered measurements, with the constraints of the governing equations. Leveraging automatic differentiation (AD) capabilities in deep learning, PINNs can compute partial derivatives accurately, enabling them to solve PDEs and identify unknown governing parameters. In contrast, automatic differentiation does not readily apply to PDEs with non-local operators, such as fractional PDEs. To address this difficulty, Pang et al. [4] extended PINNs to fPINNs for solving space-time fractional advection-diffusion equations. fPINNs innovate using a hybrid approach that combines AD for integer-order operators with classical techniques like finite difference methods for fractional-order operators.

In this paper, we introduce an enhanced version of the fractional PINNs tailored specifically for addressing fractional convection-diffusion problems. Our approach is complemented by an innovative neural network architecture designed to improve the performance of fPINNs in this context.

The structure of this paper is organized as follows. In Section 2, we provide preliminary information, including an overview of the fractional convection-diffusion problem. Section 3 highlights our key findings and introduces the innovative framework we developed. In Section 4, we detail our time discretization strategy. Section 5 presents a numerical example that demonstrates the effectiveness of our method. Finally, Sec-

*Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran., mohammadi.maryam@math.iut.ac.ir

†Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran., mohadeseh.ramezani@math.iut.ac.ir

‡Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran., mokhtari@iut.ac.ir

tion 6 concludes the paper with a summary of our main findings and discusses potential future research avenues.

2 Preliminaries

We consider the following time fractional convection-diffusion problem

$$\begin{aligned} &\mathrm{D}_t^\alpha u = \Delta u - u + f(\boldsymbol{x},t), && (\boldsymbol{x},t) \in \Omega \times (0,T], \\ &u(\boldsymbol{x},t) = g(\boldsymbol{x},t), && (\boldsymbol{x},t) \in \partial\Omega \times [0,T], \\ &u(\boldsymbol{x},t) = u_0(\boldsymbol{x}), && \boldsymbol{x} \in \Omega, \end{aligned} \tag{1}$$

on a bounded convex domain $\Omega \subset \mathbb{R}^2$. In (1), $\Delta = \partial_{x_1}^2 + \partial_{x_2}^2$ and D_t^α stands for the Caputo fractional derivative of order $\alpha \in (0,1)$, which is defined as [2]

$$\mathrm{D}_t^\alpha u(\cdot,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} \, \frac{\partial}{\partial t} u(\cdot,s) \, \mathrm{d}s. \tag{2}$$

As mentioned in [1], the solution u for problem (1) demonstrates weak singular behavior at initial time. In the upcoming sections, we will present the proposed framework to handle problem (1), incorporating the weak initial singularity property.

3 The modified fPINN framework

To solve the fractional convection-diffusion problem (1), we employ a radial basis function neural network, which uses radial basis functions as activation functions to approximate the solution. In our model, we specifically use Gaussian radial basis functions, defined as

$$\mathcal{R}_{\mathrm{g}}(\|\boldsymbol{x} - \boldsymbol{c}\|) = e^{-\|\boldsymbol{x}-\boldsymbol{c}\|^2}.$$

With the Gaussian function serving as the activation function, the relationship between the input and output in the RBF neural network can mathematically be formulated as follows

$$\tilde{u}(\boldsymbol{x}) = \sum_{i=1}^{m} w_i e^{-\|\boldsymbol{x}-\boldsymbol{c}_i\|^2},$$

Here, m indicates the total number of radial basis functions (or hidden nodes), and $\|\cdot\|$ signifies the Euclidean norm. The term w_i represents the connection weight from the i-th hidden unit to the output unit, while $\boldsymbol{c_i}$ denotes the center associated with the i-th node. We define the notation $\lambda = \{w_i, \boldsymbol{c}_i\}_{i=1}^m$ to include all trainable parameters of the network. As a result, the output of the network can be rewritten as $u_{Net}(\boldsymbol{x}, t; \lambda)$. In our framework, we can choose a form of the approximate solution that inherently satisfies both the initial and boundary conditions. Specifically, we define the approximate solution as

$$\tilde{u}(\boldsymbol{x},t) = t \; \rho(\boldsymbol{x}) \; u_{Net}(\boldsymbol{x},t;\lambda) + g(\boldsymbol{x},t),$$

where $\rho(\boldsymbol{x})$ is an auxiliary function preselected to ensure that the boundary and initial conditions are automatically satisfied.

In this scenario, we specify the operator $\mathcal{L}[u(\boldsymbol{x},t)]$ in the following manner, where $\mathcal{L}$ describes the dynamics of the underlying fractional reaction-subdiffusion problem

$$\mathcal{L}[u(\boldsymbol{x},t)] := \mathrm{D}_t^\alpha u(\boldsymbol{x},t) - \Delta u(\boldsymbol{x},t) + u(\boldsymbol{x},t).$$

We can classify the operators that comprise $\mathcal{L}$ into two distinct categories:

$$\mathcal{L} := \mathcal{L}_{\mathrm{nonAD}} + \mathcal{L}_{\mathrm{AD}} := \mathrm{D}_t^\alpha + \Delta - u.$$

The $\mathcal{L}_{\mathrm{AD}}$ component contains operators that can be evaluated using automatic differentiation. Conversely, $\mathcal{L}_{\mathrm{nonAD}}$ consists of operators that cannot be computed through automatic differentiation. For this term, numerical approximation techniques are required, which will be discussed in greater detail later.

To effectively train our model, we delineate the loss function with the following formalism

$$\begin{aligned} \mathrm{L}(\lambda) = \frac{1}{|\psi|} \sum_{(\boldsymbol{x},t)\in\psi} \Big(&\mathcal{L}_{\mathrm{nonAD}}[\tilde{u}(\boldsymbol{x},t)] + \mathcal{L}_{\mathrm{AD}}[\tilde{u}(\boldsymbol{x},t)] \\ &- f(\boldsymbol{x},t) \Big)^2, \end{aligned} \tag{3}$$

wherein, $|\psi|$ the number of training points in the training sets $\psi \subset \Omega \times (0,T]$. The training process for the proposed model in this problem involves minimizing the loss function (3) concerning λ using the L-BFGS optimizer to find the optimal parameters for the RBF network.

4 Time discretization

In this section, we concentrate on discretizing the Caputo fractional derivative (2) using the L1 formula [3]. While S-type formulas [7, 6] offer potential accuracy improvements, we opt for the L1 formula here, deferring the examination of S-type formulas for future research. For a positive N, let $\tau = \frac{T}{N}$, and we $t_k = k\tau$, $0 \le k \le N$. The L1 formula yields the following result:

$$\begin{aligned} \mathrm{D}_t^\alpha \tilde{u}(\cdot,t_n) &= \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha} \, \frac{\partial \tilde{u}}{\partial t}(\cdot,s) \; \mathrm{d}s \\ &\simeq \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \frac{\tilde{u}(\cdot,t_{k+1}) - \tilde{u}(\cdot,t_k)}{\tau} \\ &\quad \times \int_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha} \; \mathrm{d}s \\ &\simeq a_0^{(\alpha)} \tilde{u}(\cdot,t_n) - \sum_{k=1}^{n-1} (a_{n-k-1}^{(\alpha)} - a_{n-k}^{(\alpha)}) \tilde{u}(\cdot,t_k) \\ &\quad - a_{n-1}^{(\alpha)} \tilde{u}(\cdot,t_0), \end{aligned}$$

therein $a_l^{(\alpha)} = \dfrac{\tau^{-\alpha}}{\Gamma(2-\alpha)}\big((l+1)^{1-\alpha} - l^{1-\alpha}\big)$ for $l \geqslant 0$.

5 Numerical Results

This section aims to demonstrate the capability of the proposed approach in solving fractional PDEs. To achieve this, we focus on the fractional convection-diffusion problem, presented as follows

$$\begin{aligned} &\mathrm{D}_t^{\alpha} u - \Delta u + u = f(\boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \Omega \times [0, T], \\ &u(\boldsymbol{x}, 0) = \sin(\pi x_1)\sin(\pi x_2), \quad \boldsymbol{x} \in \Omega, \\ &u|_{\partial\Omega} = g(\boldsymbol{x}, t), \end{aligned}$$

In a bounded 2D rectangular domain Ω, with the exact solution provided by

$$u(\boldsymbol{x}, t) = t^{\alpha}\sin(x_1)\sin(x_2).$$

In the numerical experiment, we sample 50 training points on the boundary and 150 points within the interior of the spatial domain using Latin hypercube sampling to ensure a well-distributed set of points. For spatial testing, 15×15 uniformly nodes are used on the boundary and in the interior. In Figure 1, the distribution of training and test points is presented within the rectangular domain. The accuracy of our developed model is evaluated by calculating the relative L_2 error between the predicted solution and the exact solution

$$\text{Relative } \mathrm{L}_2 \text{ error} = \frac{\|\tilde{u}(\boldsymbol{x}, t) - u(\boldsymbol{x}, t)\|_2}{\|u(\boldsymbol{x}, t)\|_2},$$

where u_{NN} and u correspond to the predicted and exact solutions, respectively.

In our study, we adopt the Gaussian function as an activation function within the modified fPINN framework. This choice is intended to enhance the model's capability to accurately capture the dynamics of solutions for fractional PDEs. The training results, summarized in Table 1, show the relative L_2 errors for training and testing datasets across various fractional orders α and different temporal training nodes N. The results reveal a consistent decline in relative L_2 error with increasing N, reflecting enhanced accuracy for training and testing scenarios. Generally, higher fractional orders α correspond to significantly lower error rates, particularly at larger N values. In contrast, for lower fractional orders, while the errors remain relatively stable, a noticeable increase is evident at higher N. These findings underscore the effectiveness of the Gaussian activation function in achieving convergence and accuracy within the modified fPINN framework. Figures 2 through 4 illustrate the performance of our model.

Table 1: The relative L_2 errors of modified fPINN for different α

α	N	Train: L_2 error	Test: L_2 error
0.9	10	8.5691e-03	9.4007e-03
	20	5.0781e-03	5.2961e-03
	40	2.9811e-03	3.0333e-03
	80	1.8270e-03	1.8407e-03
0.5	10	2.0597e-02	3.1386e-02
	20	1.3174e-02	1.7935e-02
	40	8.3898e-03	1.0132e-02
	80	7.0808e-03	7.3436e-03
0.1	10	8.8966e-03	9.2007e-02
	20	1.8379e-02	6.5814e-02
	40	3.2803e-02	5.3172e-02
	80	4.7975e-02	5.2555e-02

6 Conclusion

We have introduced a novel fPINN framework for effectively addressing fractional reaction-diffusion problems. This innovative framework integrates RBF neural networks with fundamental physical laws, showcasing a modified approach to solving fractional PDEs. Utilizing a Gaussian basis function, we have demonstrated substantial enhancements in the context of two-dimensional domains. However, in the future, we aim to utilize S-type formulas for fractional derivatives to offer heightened flexibility and efficiency of our proposed framework.

References

[1] T. Kumari and P. Roul An efficient computational technique for solving a time-fractional reaction-subdiffusion model in 2D space. *Computers & Mathematics with Applications*, 160, pp.191-208, 2024.

[2] C. Li and F. Zeng Numerical methods for fractional calculus. *CRC Pres*, 2015.

[3] K. Oldham and J. Spanier The fractional calculus. *New York: Academic Press*, 1974.

[4] G. Pang, L. Lu and G.E. Karniadakis fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing*, 41(4), pp.A2603-A2626, 2019.

[5] M. Raissi, P. Perdikaris and G.E. Karniadakis Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378, pp.686-707, 2019.

[6] M. Ramezani, R. Mokhtari A novel high-order finite-difference method for the time-fractional diffusion equation with smooth/nonsmooth solutions. *Bulletin of the Iranian Mathematical Society*, 48(6), pp. 3987-4013, 2022.

[7] M. Ramezani, R. Mokhtari and G. Haase Some high order formulae for approximating Caputo fractional derivatives. *Applied Numerical Mathematics*, 153, pp. 300-318, 2020.

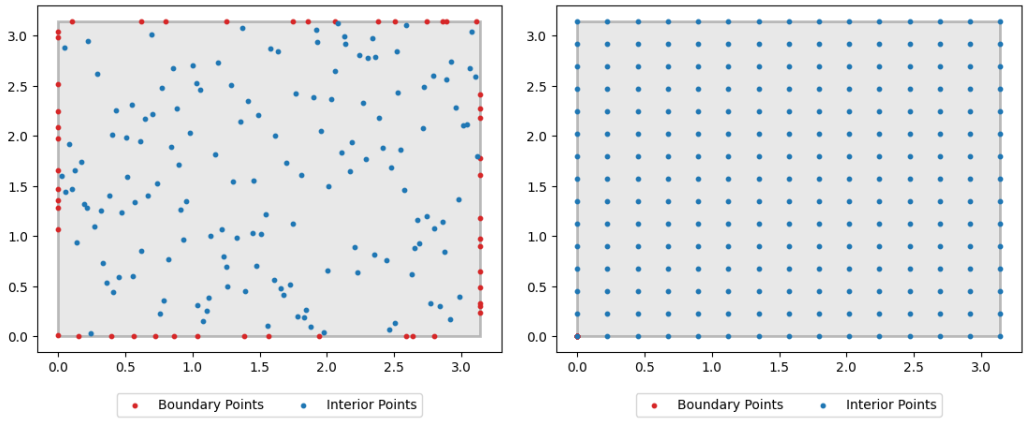

Figure 1: Data points for training (left) and testing (right).

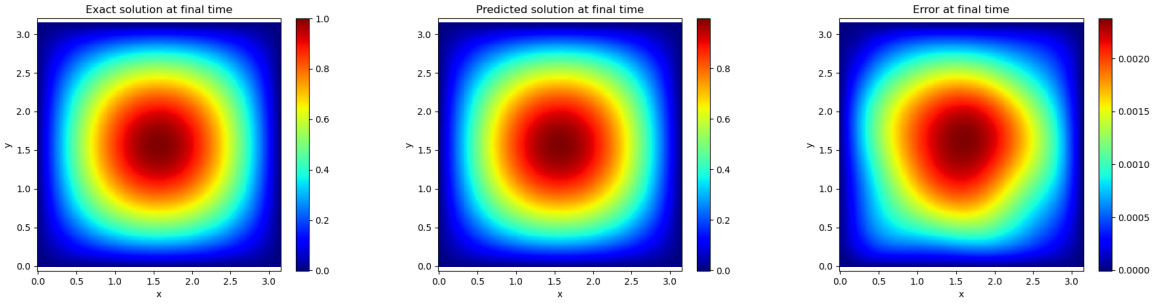

Figure 2: Exact solution, predicted solution, and error at final time for $N = 10$ and $\alpha = 0.9$.

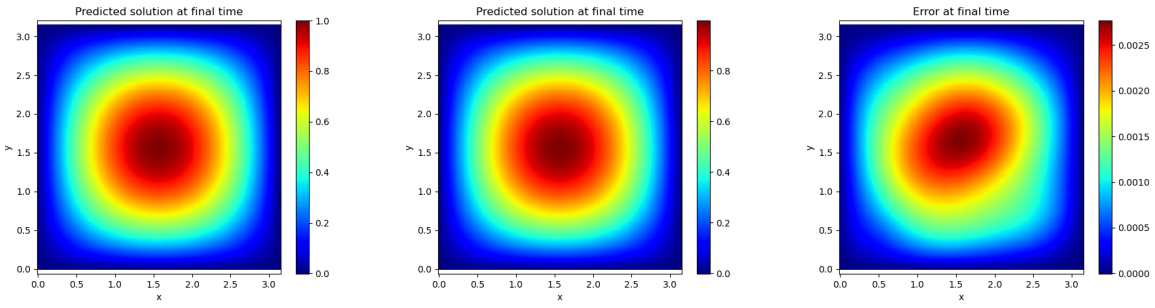

Figure 3: Exact solution, predicted solution, and error at final time for $N = 10$ and $\alpha = 0.5$.

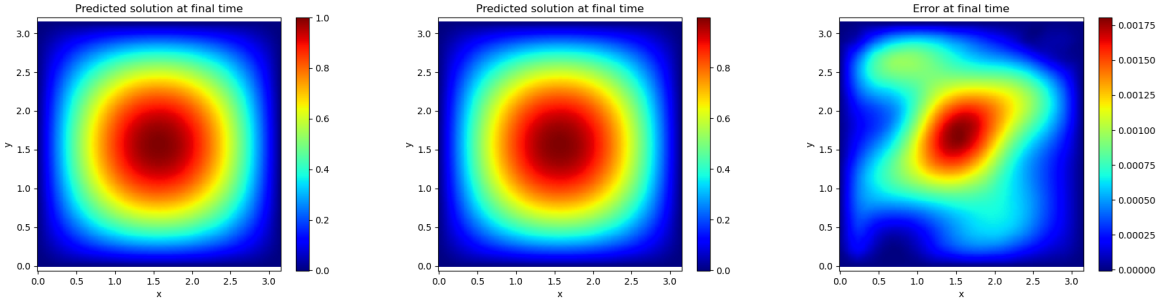

Figure 4: Exact solution, predicted solution, and error at final time for $N = 10$ and $\alpha = 0.1$.