CREAT. MATH. INFORM.
Volume **34** (2025), No. 2,
Pages 277-289

Online version at https://creative-mathematics.cunbm.utcluj.ro/
Print Edition: ISSN 1584 - 286X; Online Edition: ISSN 1843 - 441X
DOI: https://doi.org/10.37193/CMI.2025.02.10

Approximate solutions to the fractional differential equations using fractional power series approach and nature inspired optimization techniques

RATIKA RASTOGI[1], O.P. MISRA[1], AND RAJSHREE MISHRA[2]

ABSTRACT. In this paper, fractional power series (FPS) is considered as approximate function while solving fractional differential equations (FDEs). The optimal solution is then determined using metaheuristic algorithms such as particle swarm optimization (PSO) and differential evolution (DE). The optimization problems are presented using FDEs with their corresponding initial conditions. For all test problems covered in this paper, the algorithms have been developed using MATLAB software and executed on computer. A graphical comparison between the approximate solutions generated and the exact ones shows how effective the method is. Additionally, we evaluate the mean square error (MSE) between the exact and approximation solution to demonstrate the performance of our suggested technique, Differential Evolution - Fractional power series (DE-FPS) and Particle Swarm Optimization - Fractional power series (PSO-FPS), which is shown to be superior in comparison to the variational iteration method (VIM), Grey Wolf Optimization - Variational iteration method (GWO-VIM), and one other numerical iterative scheme.

1. INTRODUCTION

At present, one of the most significant areas of mathematics for simulating systems related to real-life problems is fractional calculus. It focuses heavily on the study of arbitrary order derivatives and integrals, which are important in modern mathematical research [23]. Because of its vast applicability, fractional calculus theory has been effectively applied to a variety of real-world problems during the past four decades. The fractional derivative has become more significant in problem solving in engineering and biological sciences because of the incorporation of the memory effect. Therefore, the idea of fractional integrals and derivatives has been widely used recently in a variety of biological and engineering contexts. Fractional-ordered derivatives are more widely used [37] in comparison to integer-ordered derivatives, because in most cases they offer better modeling results..

Several researchers such as Reich [32], Mateescu [22], Mastorakis [21], Babaei [5], Sadollah et. al. [34] and Rastogi et. al. [30] have used metaheuristic algorithms to solve a variety of ordinary differential equations. Also, many scholars have generally used both analytical and numerical techniques for solving the fractional-order differential equations [12, 3, 8, 36, 20, 10, 13, 1, 17, 14].

This paper presents the solution of the following type of fractional differential equations (FDEs) using fractional power series and metaheuristic algorithms:

$$D^{\alpha}u(x) = g(x, u(x)) \,, \quad n-1 < \alpha \leq n \tag{1.1}$$

Received: 27.12.2024. In revised form: 09.04.2025. Accepted: 29.05.2025
2020 *Mathematics Subject Classification.* 26A33, 34A08, 65K10.
Key words and phrases. *Fractional Power Series, Particle Swarm Optimization, Differential Evolution, Fractional Differential Equations.*
Corresponding author: Ratika Rastogi; ratikajugwalior@gmail.com

where g is a function of x and $u(x)$, and D^{α} represents the Caputo fractional derivative of order α. The Caputo FDE has a value of α between $[0, 1]$ which exists in several biological and physical systems [2, 4, 18, 19]. This type of problem was mentioned by the mathematician Leibniz in his letter to L'Hospital in 1695.

It is noted here that metaheuristic methods can operate precisely and obtain appropriate and satisfactory results related to execution time and precision, compared to classical optimization approaches, which may have problems in dealing with real-world issues due to local optimum, significant time, and difficult execution.

Differential evolution is a population-based optimization algorithm that effectively analyzes solution spaces. It is a common method for tackling optimization issues because of its uncomplicated nature and efficacy. It has been effectively employed in a variety of disciplines, including parameter estimation, data mining, and engineering design [35, 24, 27].

Particle Swarm Optimization (PSO) is a population-based optimization algorithm inspired by the collective behavior of fish schools and flock of birds. Because of its ease of use and effectiveness, it has been extensively utilized for solving optimization problems across a range of domains [9, 7, 25].

The structure of our paper is as follows. The fundamental ideas are discussed in Section 2. The terminologies related to the proposed algorithm are provided in Section 3. In Section 4, the experimental results are given. Section 5 provides the conclusion.

2. FUNDAMENTAL IDEAS [15, 16, 33, 11]

Fractional integration, fractional derivatives and fractional power series will be defined in the following subsections.

2.1. **Riemann-Liouville fractional integer** [15]**.** The Riemann-Liouville fractional integral operator of order α of a function $g(t) \in C_{\mu}, \mu \geq -1$ is $J^{\alpha}g(t) = \frac{1}{\Gamma(\alpha)}\int_0^t (t-x)^{\alpha-1}g(x)\mathrm{dx}, t > 0, J^0 g(t) = g(t)$.

For $g(t) \in C_{\mu}, \mu \geq -1, \alpha, \beta \geq 0, \gamma \geq -1$, properties of the operator J^{α}

$$J^{\alpha}J^{\beta}g(t) = J^{\beta}J^{\alpha}g(t), \quad J^{\alpha}x^{\gamma} = \frac{\Gamma(\gamma+1)}{\Gamma(\alpha+\gamma+1)}x^{\alpha+\gamma}$$

2.2. **Derivative caputo fractional** [15]**.** According to Caputo, the fractional derivative of $f(x)$ is expressed as follows:

$$D_x^{\alpha}f(x) = J^{n-\alpha}D^n f(x) = \frac{1}{\Gamma(n-\alpha)}\int_0^x (x-t)^{n-\alpha-1}f^{(n)}(t)\mathrm{dt}$$

$f(x) \in C_{\mu}^n, \mu \geq -1, \alpha, \beta \geq 0, \gamma \geq -1, n-1 < \alpha \leq n, n \in N$, properties of operator D_x^{α}

$$D_x^{\alpha}D_x^{\beta}f(x) = D_x^{\alpha+\beta}f(x) = D_x^{\beta}D_x^{\alpha}f(x),$$
$$D_x^{\alpha}x^{\gamma} = \frac{\Gamma(1+\gamma)}{\Gamma(1+\gamma-\alpha)}x^{\gamma-\alpha}, \quad x > 0$$

2.3. **Fractional Power Series** [16, 33, 11]**.** Power series is an essential tool in the analysis of elementary functions. They are commonly used in scientific computation to obtain function approximations conveniently. They have made it possible for scientists to examine various differential equations approximately in thermal physics and numerous other fields. A power series is represented as follows:

$$\sum_{n=0}^{\infty} c_n(x-x_0)^{n\alpha} = c_0 + c_1(x-x_0)^{\alpha} + c_2(x-x_0)^{2\alpha} + \ldots \tag{2.2}$$

where x_0 is the initial point, $0 \leq$ m –1 $< \alpha \leq$ m, m $\in N$ and x $\geq x_0$ is called a fractional power series (FPS) about x_0, where x is a variable and c_n are the coefficients of the series.

3. Proposed methodology for solving FDEs

The objective of this work is to illustrate a novel approach to approximately solve a variety of FDEs by using the DE or PSO technique. A variety of fundamental concepts from various disciplines of mathematics, including variational calculus, series expansion, and nature inspired optimization algorithms, are incorporated into the formulation of the suggested method. To implement an efficient problem-solving technique, each of these components would be briefly evaluated before being appropriately merged.

3.1. **Fractional differential equations.** The main goal of the strategy is to solve a variety of fractional differential equations by using the DE or PSO method with fractional power series having unknown coefficients. Instead of providing discrete numerical values at different places in the solution interval, this method gives the solution function.

The general equation for a fractional-order initial value problem can be represented as:

$$D^{\alpha}u(x) = g(x, u(x)), \tag{3.3}$$

with initial conditions

$$u^{k}(0) = b_k, \quad n-1 < \alpha \leq n \tag{3.4}$$

where k = 0,1,...,$n-1$, D^{α} represents the fractional derivative of order α, x is the independent variable, $u(x)$ is the unknown function, $g(x, u(x))$ is a given function and b_k is the value of k^{th} derivative of u at $x = 0$. The equation describes a fractional differential equation with initial conditions, where the fractional derivative captures the non-integer order behavior of the system. Solving this type of problem involves finding the function $u(x)$ that satisfies the equation and the given initial conditions.

For formulation purpose, the partial sum of the fractional power series (2.2) is considered to be the approximate solution:

$$U(x) = U_N(x) = \sum_{n=0}^{N-1} c_n (x - x_0)^{n\alpha} \tag{3.5}$$

where α is the order of fractional derivative and $c_n; n = 0, 1, 2, \ldots, N-1$ are unknown coefficients which needs to be evaluated. This function together with its caputo derivatives will be used to estimate the solution of Eq.(3.3). The total number of terms of fractional power series used in approximation is denoted by N.

It has been noted that utilizing the existing evolutionary algorithms to manage a larger set of unknown variables could potentially lead to improved accuracy. In theoretical terms, these algorithms have the capacity to handle problems with a wide range of variables. However, in practice, when confronted with a large number of variables, they often fail to identify the global optimum, as they get stuck in local optima. A broad search space need more time and computational resources for thorough analysis.

3.2. **Weighted-residual functional as a criterion for convergence** [6, 31]**.** In order to determine whether the approach has obtained the desired amount of accuracy in the approximate solution, we need an adequate criterion when using the explicit form of the fractional differential equations. To establish a criterion about the validity of the approximate solution, we require a numerical estimation of errors. If this evaluating factor is within a suitable range, we may trust the results of the algorithm. Providing an appropriate criterion as an objective function for feeding in the DE or PSO algorithm is also required. For each of these elements, we find that the concept of weighted-residual functional used in variational calculus would be appropriate. An integral that is optimally designed to assess the problem's solution is the weighted residual functional [6]. The FDE must be satisfied by the approximate solution U_N in the form of its residual-integral, which is given by

$$WRF = \int_D |W(x)| \cdot |R(x)| \, dx \tag{3.6}$$

where $W(x)$ is referred to as the weight function and $R(x)$ is the residual [31], which is obtained in the implicit form of the differential equation (3.3)

$$f(x, u(x), u^\alpha(x)) = g(x, u(x)) - D^\alpha u(x) = 0 \tag{3.7}$$

by replacing $u(x)$ and $u^\alpha(x)$ with the approximate function $U(x)$ and its fractional derivatives.

$$R(x) \equiv f\left(x, U(x), U^\alpha(x)\right) \tag{3.8}$$

WRF will be utilized as the objective function, which is being solved numerically using the Trapezoidal or Simpson rule. The weight function W(x) is an arbitrary function used in classical weighted residual methods [31]. But in the proposed method, W(x) is considered to be 1.

One of the following modifications might seem necessary throughout the solution process in order to achieve the acceptable level of precision in algorithm execution:

- In order to avoid the evolutionary mechanism from getting stuck in local optima, increase the total number of terms in fractional power series.
- Approximating the solution part by part by dividing the solution interval of the problem into small sections.

3.3. **Initial conditions formulation.** In order to solve problems related to differential equations, we need to satisfy both the equation and the equation's initial conditions. Since the nature inspired optimization process is used in the present approach to solve the differential equations, an appropriate method is required to take the IVs into consideration as optimization problem constraints. The homogeneous conditions are handled in their original implicit form as

$$u\left(x_0\right) = 0 \Rightarrow h_1\left(x_0\right) = \mid u\left(x_0\right) \mid \approx \mid U\left(x_0\right) \mid \tag{3.9a}$$

$$u'\left(x_0\right) = 0 \Rightarrow h_2\left(x_0\right) = \mid u'\left(x_0\right) \mid \approx \mid U'\left(x_0\right) \mid \tag{3.9b}$$

$$\vdots$$

$$u^{(n-1)}\left(x_0\right) = 0 \Rightarrow h_{n-1}\left(x_0\right) = \mid u^{(n-1)}\left(x_0\right) \mid \approx \mid U^{(n-1)}\left(x_0\right) \mid \tag{3.9c}$$

The constraints of the optimization problem are represented above by $h_1, h_2, \ldots, h_{n-1}$. Then, all the h_i's are included in the framework of a single penalty function explained in the next section.

FIGURE 1. The arrangement of the variables in the DE or PSO particles

3.4. **Fitness Function and Penalty Function** [29]. The DE and PSO algorithms are well known techniques for handling the unconstrained optimization problems. Consequently, in the case of constrained problems, one of the standard techniques for managing constraints is required. Therefore, the penalty function concept is used to implement the constraints discussed in the preceding section. The penalty function is in charge of penalizing solutions that violate the specified criteria. Thus, we may compute the problem's fitness function by adding together the weighted-residual integral WRF with the penalty function PF, i.e.,

$$\mathrm{FF} = \mathrm{WRF} + \mathrm{PF} \tag{3.10}$$

where the numerical value of the penalty function PF is determined using the penalty framework suggested by [29].

$$\mathrm{PF} = \mathrm{WRF} \cdot \sum_{\mathrm{i}=1}^{\mathrm{n_{IVs}}} \mathrm{K_i h_i} \tag{3.11}$$

where n_{IVs} is the number of initial conditions, and h_i is the normalised violation of the i^{th} constraint, which is obtained using equations '(3.9)' depending on the degree of violation for that current condition. The significance given to satisfying each of the criteria determines the penalty multiplier, K_i. Since choosing a large value for the constant K_i will put a lot of emphasis on satisfying this criterion, the optimization algorithms look for this constraint instead of fulfilling the differential equation. In contrast, the associated criterion will only be partially fulfilled if K_i is having a low value. Therefore, selecting appropriate values for these multipliers and adjusting them while the evolutionary process advances has its own importance, which is out of the scope of this work. However, the multipliers K_i are all assumed to remain constant throughout all examples considered here in order to keep things simple.

4. Illustrative examples

Up till now, we have discussed the flexibility of a method for determining the approximate solution of FDEs. This approach will be employed to examine several initial value problems to evaluate its relevance and accuracy. Selected examples are taken from a variety of the most recognized sources [12, 17, 19, 28] in this field. The efficiency of the algorithm is illustrated through a graphical comparison of the calculated approximations against the exact values.

To conduct a thorough analysis and comparison of the numerical results, we primarily focus on the Mean Square Error (MSE) between the exact solutions and the approximate solutions, as it serves as the key evaluation criterion in this study. The MSE is calculated using the following expression:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left\| U_N\left(t_i\right) - U_{\text{exact}}\left(t_i\right) \right\|^2 \tag{4.12}$$

The mean and variance values, which demonstrate the average precision and stability of these comparing methods, constitute the MSE values.

20 separate runs were performed for each test problem in order to complete the optimization task. The proposed method was performed on an Intel(R) Core(TM) i3 processor running at 1.70 GHz with 4 GB of RAM using the MATLAB programming software (MATLAB 2021).

The chosen values for parameters of DE algorithm is mentioned in Table 1.

TABLE 1. Parameters of DE algorithm for DE-FPS method

Parameters	Ex. 1	Ex. 2	Ex. 3
X_{min}	-1	-0.6523527	-10
X_{max}	1	0.6523527	10
$maxFE$ [a]	100000	100000	100000
M [b]	50	10	50
CP	0.9	0.9	0.9
S	0.65	0.55	0.65
It_{max} [c]	500	20	500
K_1 [d]	10	1000	100000

[a] Maximum Function Evaluations
[b] Population Size
[c] Maximum Iterations
[d] Penalty Multipliers

The chosen values for parameters of PSO algorithm is mentioned in Table 2.

TABLE 2. Parameters of PSO algorithm for PSO-FPS method

Parameters	Ex. 1	Ex. 2	Ex. 3
X_{min}	-2	-0.6523527	-2
X_{max}	2	0.6523527	2
$maxFE$ [a]	100000	100000	100000
M [b]	50	10	50
s_1	0.5	0.5	0.5
s_2	1.5	1.5	1.5
ω_{min}	0.4	0.4	0.3
ω_{max}	0.9	0.9	0.8
It_{max} [c]	500	20	200
K_1 [d]	100	1000	100000

[a] Maximum Function Evaluations
[b] Population Size
[c] Maximum Iterations
[d] Penalty Multipliers

4.1. **Example 1** [19]**.** Consider the following fractional differential equation

$$D^{\alpha}u(x) + u(x) = x^2 + \frac{2x^{1.5}}{\Gamma(2.5)},\ 0 \le x \le 1,\ 0 < \alpha \le 1 \tag{4.13}$$

$$u(0) = 0 \tag{4.14}$$

The exact solution for (4.13) and (4.14) is

$$u_{\text{exact}}\,(x) = x^2 \; when \; \alpha = 0.5 \tag{4.15}$$

The best approximate solution to this problem using Differential Evolution method and Fractional power series (DE-FPS) is obtained for N $= 10$. Thus, we get the approximate solution as

$$u(x) \approx 7.687 \times 10^{-5}\, x^{0.5} - 1.896 \times 10^{-3}\, x + 2.054 \times 10^{-2}\, x^{1.5} + 0.8905\, x^2 \\ +0.3183\, x^{2.5} - 0.531\, x^3 + 0.5088\, x^{3.5} - 0.2604\, x^4 + 5.517 \times 10^{-2}\, x^{4.5}$$

The best approximate solution to this problem using Particle Swarm Optimization method and Fractional power series (PSO-FPS) is obtained for N = 10. Thus, we get the approximate solution as

$$u(x) \approx -8.618 \times 10^{-8} + 7.36 \times 10^{-2}\, x^{0.5} - 0.6623\, x + 2\, x^{1.5} - 1.4039\, x^2 \\ +0.8882\, x^{2.5} + 0.2293\, x^3 - 0.6848\, x^{3.5} + 1.2677\, x^4 - 0.7093\, x^{4.5}$$

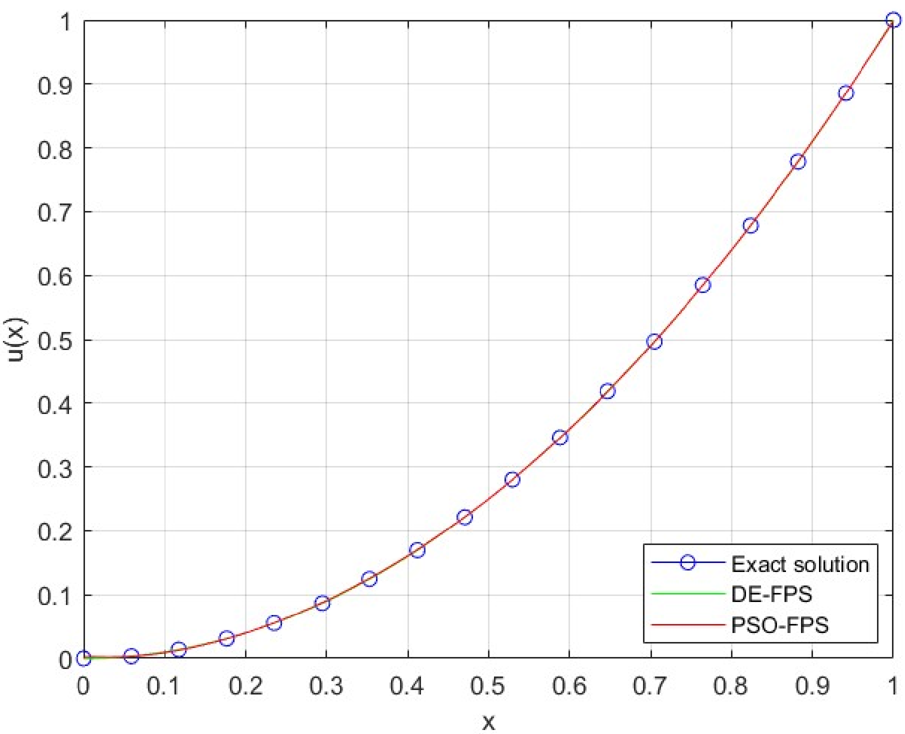

FIGURE 2. Comparison of the exact and approximate solutions of Example 1 using DE-FPS and PSO-FPS methods

The graphical comparison between the exact and approximate solutions of Example 1 using the DE-FPS and PSO-FPS methods is depicted in Fig. 2. It is clear from the graph that both the DE-FPS and PSO-FPS methods detected the exact solution with adequate precision.

4.2. **Example 2** [28]. Consider the nonlinear Riccati differential equation

(4.16) $$D^{\alpha} u(x) + u^2(x) = 1,\ 0 \leq x \leq 1,\ 0 < \alpha \leq 1$$

(4.17) $$u(0) = 0$$

TABLE 3. Comparison table for MSE of the solution of Example 1 obtained by the proposed DE-FPS and PSO-FPS methods with that of other techniques [17]

Technique	(MSE)
DE-FPS	3.8093×10^{-12}
PSO-FPS	4.4541×10^{-7}
VIM [17]	2.6283
GWO-VIM [17]	2.600×10^{-3}

The exact solution for (4.16) and (4.17) is

$$u_{\text{exact}}(x) = \frac{e^{2x}-1}{e^{2x}+1} \; when \; \alpha = 0.75 \tag{4.18}$$

The best approximate solution to this problem using Differential Evolution method and Fractional power series (DE-FPS) is obtained for N = 20. Thus, we get the approximate solution as

$$\begin{gathered} u(x) \approx \; 0.3824\, x^{0.75} + 0.6524\, x^{1.5} + 0.6524\, x^{2.25} - 0.6524\, x^{3} - 0.4151\, x^{3.75} \\ +0.2936\, x^{4.5} - 0.4861\, x^{5.25} - 0.6524\, x^{6} + 0.3836\, x^{6.75} - 0.2059\, x^{7.5} \\ +0.6524\, x^{8.25} + 3.869 \times 10^{-2}\, x^{9} - 0.1973\, x^{9.75} + 0.4689\, x^{10.5} + 0.5561\, x^{11.25} \\ -0.2579\, x^{12} + 0.5636\, x^{12.75} - 0.6524\, x^{13.5} - 0.3836\, x^{14.25} \end{gathered}$$

The best approximate solution to this problem using Particle Swarm Optimization method and Fractional power series (PSO-FPS) is obtained for N = 20. Thus, we get the approximate solution as

$$\begin{gathered} u(x) \approx \; 0.6524\, x^{0.75} + 2.587 \times 10^{-2}\, x^{1.5} + 0.6298\, x^{2.25} - 4.614 \times 10^{-2}\, x^{3} \\ -0.6524\, x^{3.75} - 8.007 \times 10^{-2}\, x^{5.25} + 9.838 \times 10^{-3}\, x^{6.75} + 0.6524\, x^{9} \\ -0.6524\, x^{11.25} - 0.6524\, x^{12} - 4.151 \times 10^{-2}\, x^{12.75} + 0.6524\, x^{13.5} \\ +0.2397\, x^{14.25} \end{gathered}$$

The graphical comparison between the exact and approximate solutions of Example 2 using the DE-FPS and PSO-FPS methods is depicted in Fig. 3. It is seen from the graph that both the DE-FPS and PSO-FPS methods detected the exact solution with adequate precision, but the approximate solution obtained by DE-FPS method in this case is better than that obtained by PSO-FPS method as far as accuracy is concerned.

TABLE 4. Comparison table for MSE of the solution of Example 2 obtained by the proposed DE-FPS and PSO-FPS methods with that of other techniques [17]

Technique	(MSE)
DE-FPS	1.6778×10^{-4}
PSO-FPS	3.8067×10^{-4}
VIM [17]	4.3000×10^{-3}
GWO-VIM [17]	1.3000×10^{-3}

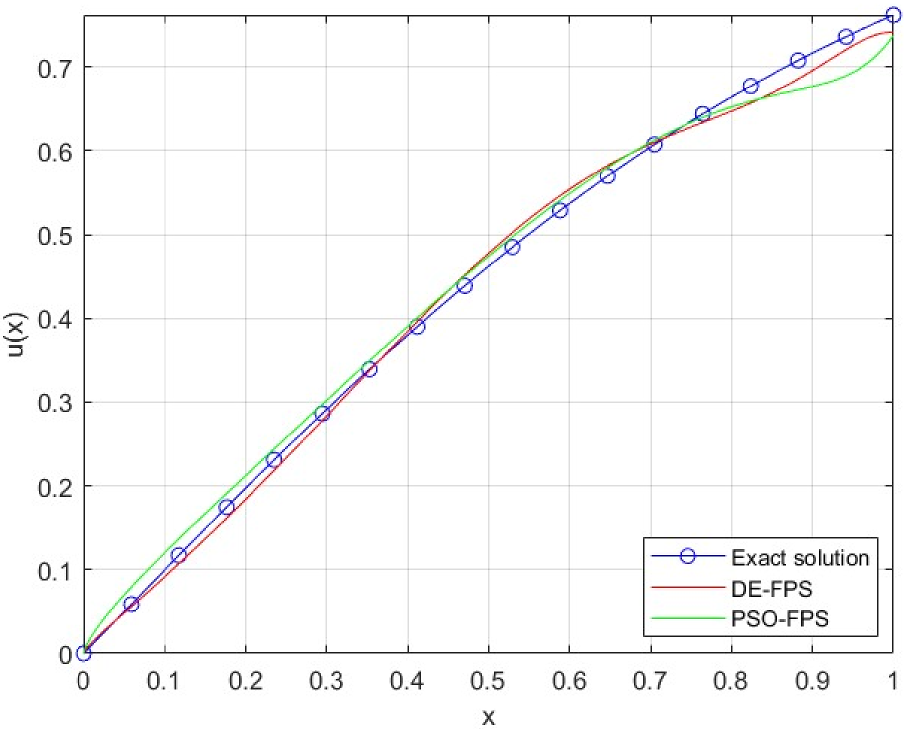

FIGURE 3. Comparison of the exact and approximate solutions of Example 2 using DE-FPS and PSO-FPS methods

4.3. **Example 3** [12]. Consider the example of nonlinear fractional differential equation which is used to solve an initial value problem describing the process of cooling of a semi-infinite body by radiation

$$D^{\alpha}u(x) - \gamma(p_0 - u(x))^4 = 1,\ 0 \leq x \leq 1,\ 0 < \alpha \leq 1 \tag{4.19}$$

$$u(0) = 0 \tag{4.20}$$

The exact solution for (4.19) and (4.20) is

$$u_{\text{exact}}(x) = p_0 - \left(\frac{p_0^3\sqrt{\pi}}{6\sqrt{x} + \sqrt{\pi}}\right)^{1/3} \; when\ \alpha = 0.5 \tag{4.21}$$

For γ = 1 and p_0 = 1, the best approximate solution to this problem using Differential Evolution method and Fractional power series (DE-FPS) is obtained for N = 10. Thus, we get the approximate solution as

$$u(x) \approx \; 1.1261\, x^{0.5} - 3.1629\, x + 6.8434\, x^{1.5} - 8.999\, x^2 + 5.8534\, x^{2.5}$$
$$-0.2693\, x^3 - 1.7134\, x^{3.5} + 0.5794\, x^4 + 5.587 \times 10^{-2}\, x^{4.5}$$

For γ = 1 and p_0 = 1, the best approximate solution to this problem using Particle Swarm Optimization method and Fractional power series (PSO-FPS) is obtained for N = 10. Thus, we get the approximate solution as

$$u(x) \approx 1.1785\,x^{0.5} - 2\,x + 0.242\,x^{1.5} + 1.8636\,x^2 + 1.0491\,x^{2.5}$$
$$-1.7924\,x^3 - 2\,x^{3.5} + 1.3669\,x^4 + 0.416\,x^{4.5}$$

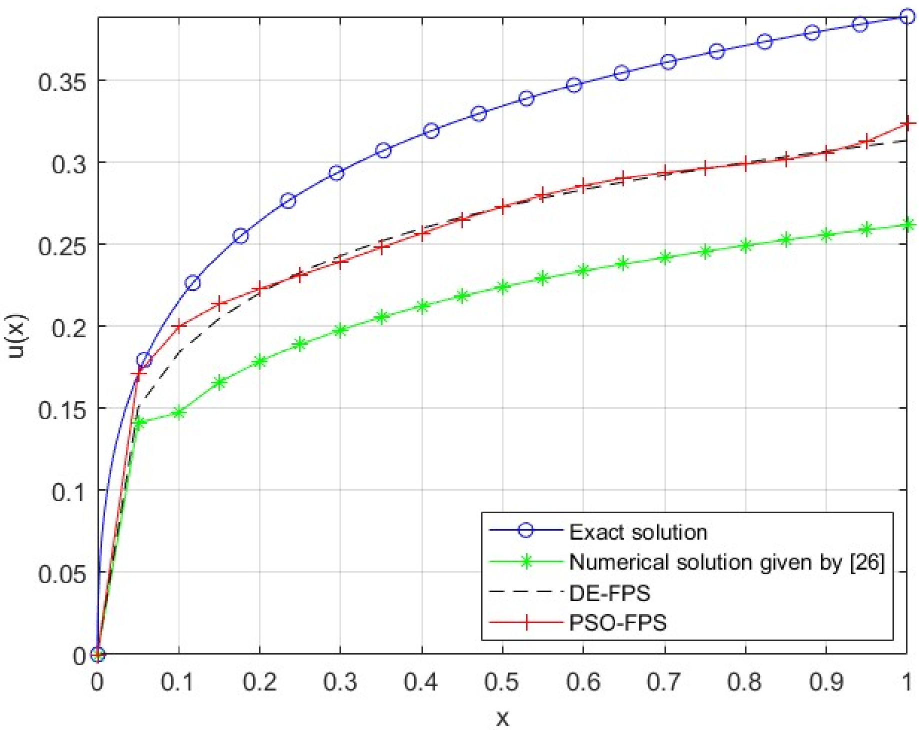

FIGURE 4. Comparison of the exact and approximate solutions of Example 3 using DE-FPS and PSO-FPS methods

The graphical comparison between the exact and approximate solutions of Example 3 using the DE-FPS and PSO-FPS methods is depicted in Fig. 4. It is seen that both the DE-FPS and PSO-FPS methods outperformed the other method in terms of accuracy. Also, the approximate solution obtained by PSO-FPS method in this case is better than that obtained by DE-FPS method as far as accuracy is concerned.

TABLE 5. Comparison table for MSE of the solution of Example 3 obtained by the proposed DE-FPS and PSO-FPS methods with that of other numerical technique [26]

Technique	(MSE)
DE-FPS	3.4715×10^{-3}
PSO-FPS	3.3338×10^{-3}
Numerical method mentioned by [26] for h = 0.05	1.0827×10^{-2}

TABLE 6. Mean Function Evaluations (Mean FE) for the solutions to all the three examples obtained by the proposed DE-FPS and PSO-FPS methods

Mean FE	Ex. 1	Ex. 2	Ex. 3
DE-FPS	1337.5	82	14172.5
PSO-FPS	6102.5	103.5	17817.5

5. Conclusion

In this paper, fractional differential equations are solved numerically to get their approximate solutions. For this, fractional power series with unknown coefficients are taken as base approximation function and then these coefficients are evaluated using meta-heuristic optimization techniques such as Differential Evolution (DE) and Particle Swarm Optimization (PSO) in order to achieve desired precision. Three problems are considered as examples to illustrate our proposed method. The MSE of the approximate solutions for Ex. 1 derived from both DE-FPS and PSO-FPS methods w.r.t. exact solutions are superior to that of VIM and GWO-VIM methods which were developed by Entesar and Qasim [17] (See Fig. 2 and Table 3). Further, it may be noted that our DE-FPS method for this example is even better than that of our PSO-FPS method.

It is further noted that the MSE of the approximate solutions for Ex. 2 obtained by both DE-FPS and PSO-FPS methods w.r.t. exact solutions are better than the MSE of the solutions derived from VIM and GWO-VIM methods which were suggested by Entesar and Qasim [17] (See Fig. 3 and Table 4). Here, it may be noted that the DE-FPS method for this example is still better than the PSO-FPS method.

The MSE for the approximate solutions for Ex. 3 obtained by both DE-FPS and PSO-FPS methods w.r.t. exact solutions are better than that of the MSE of the solution obtained by the numerical method as suggested in Podlubny [26] (See Fig. 4 and Table 5). Further, it is noted that the PSO-FPS method in this case is even better than the DE-FPS method.

It is seen from Table 6 that the mean function evaluations (Mean FE) for DE-FPS method is less than that of PSO-FPS method in case of all the three examples 1, 2 and 3 considered in this paper. It can be concluded here that the DE-FPS method is more efficient than PSO-FPS method as far as mean FE is concerned.

After observing the outcomes of the proposed method in this paper, we may conclude that the performance of both the DE-FPS and PSO-FPS methods are much better than the methods suggested earlier by Entesar and Qasim [17] and Podlubny [26].

References

[1] Ahmad, A.; Sulaiman, M.; Kumam, P. Solutions of fractional order differential equations modeling temperature distribution in convective straight fins design, *Adv. Differ. Equ.* **2021** (2021), no. 1, 1–38.

[2] Ahmed, E.; El-Sayed, A.; El-Saka, H.,A. Equilibrium points, stability and numerical solutions of fractional-order predator-prey and rabies models, *J. Math. Anal. Appl.* **325** (2007), no. 1, 542–553.

[3] Ali, K. K.; Abd El Salam, M. A.; Mohamed, E. M.; Samet, B.; Kumar, S.; Osman, M. S. Numerical solution for generalized nonlinear fractional integro-differential equations with linear functional arguments using Chebyshev series, *Adv. Differ. Equ.* **2020** (2020), no. 1, 1–23.

[4] Al-Talib, Z. S.; Al-Azzawi, S. F. Projective and hybrid projective synchronization of 4-D hyperchaotic system via nonlinear controller strategy, *Telkomnika* **18** (2020), no. 2, 1012–1020.

[5] Babaei, M. A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization, *Appl. Soft Comput.* **13** (2013), no. 7, 3354–3365.

[6] Bathe, K.J. *Finite Element Procedures*, 2nd ed., Prentice Hall, New Jersey, 1996.

[7] Bansal, J.C.; Deep, K. A modified binary particle swarm optimization for knapsack problems, *Appl Math Comput.* **218** (2012), no. 22, 11042–11061.

[8] Bukhari, A. H.; Raja, M. A. Z.; Sulaiman, M.; Islam, S.; Shoaib, M.; Kumam, P. Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting, *IEEE Access 8*, (2020), 71326–71338.

[9] Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *TEVC* **6** (2002), no. 1, 58–73.

[10] Cuahutenango-Barro, B.; Taneco-Hernández, M.; Lv, Y. P.; Gómez-Aguilar, J.; Osman, M.; Jahanshahi, H.; Aly, A. A. Analytical solutions of fractional wave equation with memory effect using the fractional derivative with exponential kernel, *Results Phys.* **25** (2021), 104148.

[11] Cui, R.; Hu, Y. Fractional power series method for solving fractional differemtial equation, *J. adv. math.* **14** (2016), no. 4, 6156–6159.

[12] Demirci, E.; Ozalp, N. A method for solving differential equations of fractional order, *J. Comput. Appl. Math.* **236** (2012), no. 11, 2754–2762.

[13] Djennadi, S.; Shawagfeh, N.; Osman, M. S.; Gómez-Aguilar, J. F.; Arqub, O. A. The Tikhonov regularization method for the inverse source problem of time fractional heat equation in the view of ABC-fractional technique, *Phys. Scr.* **96** (2021), no. 9, 094006.

[14] Ebrahimzadeh, A.; Khanduzi, R.; A Beik, S. P.; Baleanu, D. Research on a collocation approach and three metaheuristic techniques based on MVO, MFO, and WOA for optimal control of fractional differential equation, *J. Vib. Control.* **29** (2023), no. 3-4, 661–674.

[15] El-Ajou, A.; Odibat, Z.; Momani, S.; Alawneh, A. Construction of analytical solutions to fractional differential equations using homotopy analysis method, *IAENG Int. J. Appl. Math.* **40** (2010), no. 2.

[16] El-Ajou, A.; Arqub, O. A.; Zhour, Z. A.; Momani, S. New results on fractional power series: theories and applications, *Entropy* **15** (2013), no. 12, 5305–5323.

[17] Entesar, A.; Qasim, O. S. Solve fractional differential equations via a hybrid method between variational iteration method and gray wolf optimization algorithm, *Asian-Eur. J. Math.* **14** (2021), no. 8, 2150144.

[18] Glöckle, W. G.; Nonnenmacher, T. F. A fractional calculus approach to self-similar protein dynamics, *Biophys. J.* **68** (1995), no. 1, 46–53.

[19] Kilbas, A. A.; Srivastava, H. M.; Trujillo, J. J. Theory and Applications of Fractional Differential Equations, *J. Electrochem. Soc.* **204** (2006).

[20] Kumar, S.; Kumar, R.; Osman, M. S.; Samet, B. A wavelet based numerical scheme for fractional order SEIR epidemic of measles by using Genocchi polynomials, *Numer. Methods Partial Differ. Equ.* **37** (2021), no. 2, 1250–1268.

[21] Mastorakis, E. N. Unstable ordinary differential equations: solution via genetic algorithms and the method of Nelder-Mead, *Proceedings of the 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation* **119**, Elounda, Greece, (2007), 297–354.

[22] Mateescu, G. D. On the application of genetic algorithms to differential equations, *Rom. J. Econ. Forecast.* **7** (2006), no. 2, 5–9.

[23] Miller, K.S.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*, Wiley, New York, 1993.

[24] Pant, M.; Zaheer, H.; Garcia-Hernandez, L.; Abraham, A. Differential Evolution: A review of more than two decades of research, *Eng. Appl. Artif.* **90** (2020), 103479.

[25] Papadimitrakis, M.; Kapnopoulos, A.; Tsavartzidis, S.; Alexandridis, A. A cooperative PSO algorithm for Volt-VAR optimization in smart distribution grids, *Electr. Power Syst. Res.* **212** (2022), 108618.

[26] Podlubny, I. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Elsevier, 1998.

[27] Praczyk, T. Using Hill Climb Modular Assembler Encoding and Differential Evolution to evolve modular neuro-controllers of an autonomous underwater vehicle acting as a Magnetic Anomaly Detector, *Appl. Soft Comput.*, (2022), 109347.

[28] Pradhan, M.; Roy, P. K.; Pal, T. Grey wolf optimization applied to economic load dispatch problems, *Int. J. Electr. Power Energy Syst.* **83** (2016), 325–334.

[29] Rajeev, S.; Krishnamoorthy, C. S. Discrete optimization of structures using genetic algorithms, *J. Struct. Eng.* **118** (1992), no. 5, 1233–1250.

[30] Rastogi, R.; Misra, O. P.; Mishra, R. A Chebyshev polynomial approach to approximate solution of differential equations using differential evolution, *Eng. Appl. Artif.* **126** (2023), 107197.

[31] Reddy, J. N. *Introduction to the finite element method*, McGraw-Hill Education, 2019.

[32] Reich, C. Simulation of imprecise ordinary differential equations using evolutionary algorithms, *Proceedings of the 2000 ACM Symposium on Applied Computing* **1** (2000), 428–432.

[33] Ren, F.; Hu, Y. The fractional power series method an efficient candidate for solving fractional systems, *Therm. Sci.* **22** (2018), no. 4, 1745–1751.

[34] Sadollah, A.; Eskandar, H.; Kim, J. H. Approximate solving of nonlinear ordinary differential equations using least square weight function and metaheuristic algorithms, *Eng. Appl. Artif.* **40** (2015), 117–132.

[35] Sharma, H.; Bansal, J. C.; Arya, K. V. Fitness based differential evolution, *Memetic Comput.* **4** (2012), no. 4, 303–316.

[36] Waseem, W.; Sulaiman, M.; Alhindi, A.; Alhakami, H. A soft computing approach based on fractional order DPSO algorithm designed to solve the corneal model for eye surgery, *IEEE Access 8* (2020), 61576—61592.

[37] Zhou, P.; Kuang, F. A novel control method for integer orders chaos systems via fractional-order derivative, *Discrete Dyn. Nature Soc.* **2011** (2011), 1—8.

[1] School of Mathematics and Allied Sciences, Jiwaji University, Gwalior, 474011, Madhya Pradesh, India

Email address: ratikajugwalior@gmail.com, misraop09@gmail.com

[2] Department Of Mathematics, Shrimant Madhavrao Scindia Government Model Science College, Gwalior, 474009, Madhya Pradesh, India

Email address: rajshreemishraa@gmail.com