

## RESEARCH ARTICLE

## Investigation of singular ordinary differential equations by a neuroevolutionary approach

Waseem Waseem<sup>1</sup>, Muhammad Sulaiman<sup>1\*</sup>, Poom Kumam<sup>2,3,4\*</sup>, Muhammad Shoaib<sup>7</sup>, Muhammad Asif Zahoor Raja<sup>5,6</sup>, Saeed Islam<sup>1,8,9</sup>

**1** Department of Mathematics, Abdul Wali Khan University Mardan, KP, Pakistan, **2** KMUTT Fixed Point Research Laboratory, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand, **3** KMUTT-Fixed Point Theory and Applications Research Group, Theoretical and Computational Science Center (TaCS), Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand, **4** Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan, **5** Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan, R.O.C., **6** Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock, Pakistan, **7** Department of Mathematics, COMSATS University Islamabad, Attock, Pakistan, **8** Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam, **9** Faculty of Mathematics & Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam

\* [msulaiman@awkum.edu.pk](mailto:msulaiman@awkum.edu.pk) (MS); [poom.kum@kmutt.ac.th](mailto:poom.kum@kmutt.ac.th) (PK)

## OPEN ACCESS

**Citation:** Waseem W, Sulaiman M, Kumam P, Shoaib M, Raja MAZ, Islam S (2020) Investigation of singular ordinary differential equations by a neuroevolutionary approach. PLoS ONE 15(7): e0235829. <https://doi.org/10.1371/journal.pone.0235829>

**Editor:** Hector Vazquez-Leal, Universidad Veracruzana, MEXICO

**Received:** February 1, 2020

**Accepted:** June 23, 2020

**Published:** July 9, 2020

**Copyright:** © 2020 Waseem et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are within the paper.

**Funding:** The authors acknowledge the financial support provided by the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), King Mongkut's University of Technology Thonburi (KMUTT). Funder role: Professor Poom Kumam had a role in the preparation of the manuscript, and financial support in article processing charges.

## Abstract

In this research, we have investigated doubly singular ordinary differential equations and a real application problem of studying the temperature profile in a porous fin model. We have suggested a novel soft computing strategy for the training of unknown weights involved in the feed-forward artificial neural networks (ANNs). Our neuroevolutionary approach is used to suggest approximate solutions to a highly nonlinear doubly singular type of differential equations. We have considered a real application from thermodynamics, which analyses the temperature profile in porous fins. For this purpose, we have used the optimizer, namely, the fractional-order particle swarm optimization technique (FO-DPSO), to minimize errors in solutions through fitness functions. ANNs are used to design the approximate series of solutions to problems considered in this paper. We find the values of unknown weights such that the approximate solutions to these problems have a minimum residual error. For global search in the domain, we have initialized FO-DPSO with random solutions, and it collects best so far solutions in each generation/ iteration. In the second phase, we have fine-tuned our algorithm by initializing FO-DPSO with the collection of best so far solutions. It is graphically illustrated that this strategy is very efficient in terms of convergence and minimum mean squared error in its best solutions. We can use this strategy for the higher-order system of differential equations modeling different important real applications.

## 1 Introduction

Real-world problems which are modeled as a singular boundary value problem (BVP) of ordinary differential equations are often hard to solve. Such systems frequently arise in physics, or

**Competing interests:** The authors have declared that no competing interests exist.

specifically astrophysics, thermodynamics, physical chemistry, nuclear technology, atomic energy, and all studies involving non-linear conic systems [1–4]. Singular BVPs are been tackled numerically and analytically by different researchers using techniques like the monotonic iterative method of Bessel functions [5], an improved iterative technique [6], homotopy perturbation method (HPM) [7], finite difference method with uniform mesh [8], monotonic iterative technique involving expansion of eigenfunction [9], modified adomian decomposition method (MADM) [10], Borel–Laplace transformation technique [11], and approximate power series solution method [12].

A review of all these numerical methods shows that they are deterministic and require prior information about the problem. Which is a disadvantage in case we do not have any information about a problem under consideration [13–16]. One example of such problem is the class of doubly non-linear singular differential equations. Meta-heuristic techniques are better alternatives for a variety of singular differential equations like doubly non-linear singular problems.

In this study, a soft computing approach based on hybridization of feed-forward neural networks and fractional-order darwinian PSO is suggested. To investigate the capability of our approach, we have solved three singular non-linear differential equation known as differential equations with doubly singularities. To further analyse our approach, we have solved nine sub-cases with different combinations of parameters. A real application is also considered in problem 4 to further highlight the effectiveness of our approach. We give a general representation of this system in Eq (1).

$$(p(x)y'(x))' = q(x)f(x, y(x)), \quad 0 < x \leq 1, \quad (1)$$

this differential equation is subject to the Dirichlet type of boundary conditions, which are given below.

$$y(0) = a_1, \quad y(1) = c_1, \quad (2)$$

some problems are also bounded by mixed boundary conditions as

$$y(0) = 0, \quad ay(1) + by'(1) = c, \quad (3)$$

here  $a, a_1, c_1$  are non-zero positive real numbers and  $b \geq 0$ . On the other hand,  $c$  can be any real number. When the value of  $p(0)$  is zero, the system becomes a singular differential equation. If  $q(x)$  is treated as a discontinuous function over the  $y$ -axis, then the problem stated in Eqs (1)–(3) becomes a doubly-singular type of differential equation.

From the above discussion, we have understood the singular doubly boundary value problem and based on this understanding we have developed our proposed soft computing approach to get better numerical solutions of these problems.

In the recent couple of years, alternate approaches based on artificial neural networks combined with heuristics and meta-heuristic are extensively developed to solve non-linear differential equations. Some important problems which are worth mentioning, include conduction problem in electrical engineering [17, 18], thermodynamics [19, 20], non-linear pantograph differential equations [21], models of atom known as Thomas-Fermi equations [22], Fuzzy logic based problems [23], Navier-stokes equations [23], Volterra differential equations [24, 25], problems in nanofluids [26], Fredholm integro-differential equations [27], non-linear Flierl-Petviashvili differential systems [28], problems in fractional control theory [29], bilinear programming differential systems [30], flow studies of non-linear differential system of Jeffery-Hamel problems [28], Bratu differential systems [31], differential systems in electromagnetism [32].

In [33], two techniques namely GA and SQP are combined to tackle the doubly non-linear singular differential equations. However, this combined algorithm takes more time and is computationally expensive. Also, these techniques are local search routines which stuck in a local minimum. By viewing all these contributions, it has led us to design an easy-to-use approach based on soft computing, which can produce better solutions with less computational time in solving these problems which are already handled by classical techniques. The main disadvantage of those classical techniques was their requirement of prior information about the problem in hand. In this paper, our proposed approach is used to solve a second-order non-linear differential equation with double singularities and complex boundary conditions.

Our approach aims to train the unknown weights in ANN by minimizing the error function through a well-balanced single meta-heuristic algorithm known as FO-DPSO [34, 35]. We have considered different case studies of non-linear doubly singular BVPs to check the capabilities of our approach. To examine the robustness of our approach, we have performed multiple simulations to get the best values of unknown weights. A real application is considered in problem 4 to further illustrate the effectiveness of our algorithm.

Key contributions in this paper are given below:

- A theoretical and graphical model that explains why our soft computing approach works, and it is novel, is presented in section 2 (2.1, 2.2) and Figs (1), (2) and (3).
- Series solutions based on artificial neural networks are designed with the help of fractional-order particle swarm optimizer (FO-DPSO). Our novel soft computing approach is used to solve non-linear doubly singular differential equations, see Fig 1.
- A real application problem is considered to further elaborate on the competitiveness of our approach. In this problem, we have analyzed the temperature profile in a porous fin model, see Fig 8.
- We have compared our results with GA and a variant GA-SQP algorithm.
- The statistical analysis is presented in terms of absolute errors, global mean absolute error (*GMAE*), mean absolute error (*MAE*), and mean value of fitness ( $M_{fit}$ ).
- Computational times, maximum iterations took to solve our problems by ANN-based FO-DPSO are presented.
- Frequency plots of performance indicators fitted with normal distribution are graphically illustrated.

The rest of this paper is organized as, in section 2, mathematical modeling of approximate solution based on ANNs is illustrated. Fitness functions and FO-DPSO is briefly explained. Sections 3 and 4 contain problems description, numerical results for different case studies. Section 5, comprises the statistical analysis based on different performance indicators. Conclusions and future work are given in section 6.

## 2 The hybrid ANN and FO-DPSO approach

In this section, we have presented our novel approach. This approach constructs ANN-based approximate solutions with unknown weights for the doubly singular BVPs. The unknown weights are determined such that the approximate solution satisfies the problem with a less residual error. We have presented a detailed graphical abstract of the novel procedure in Fig 1.

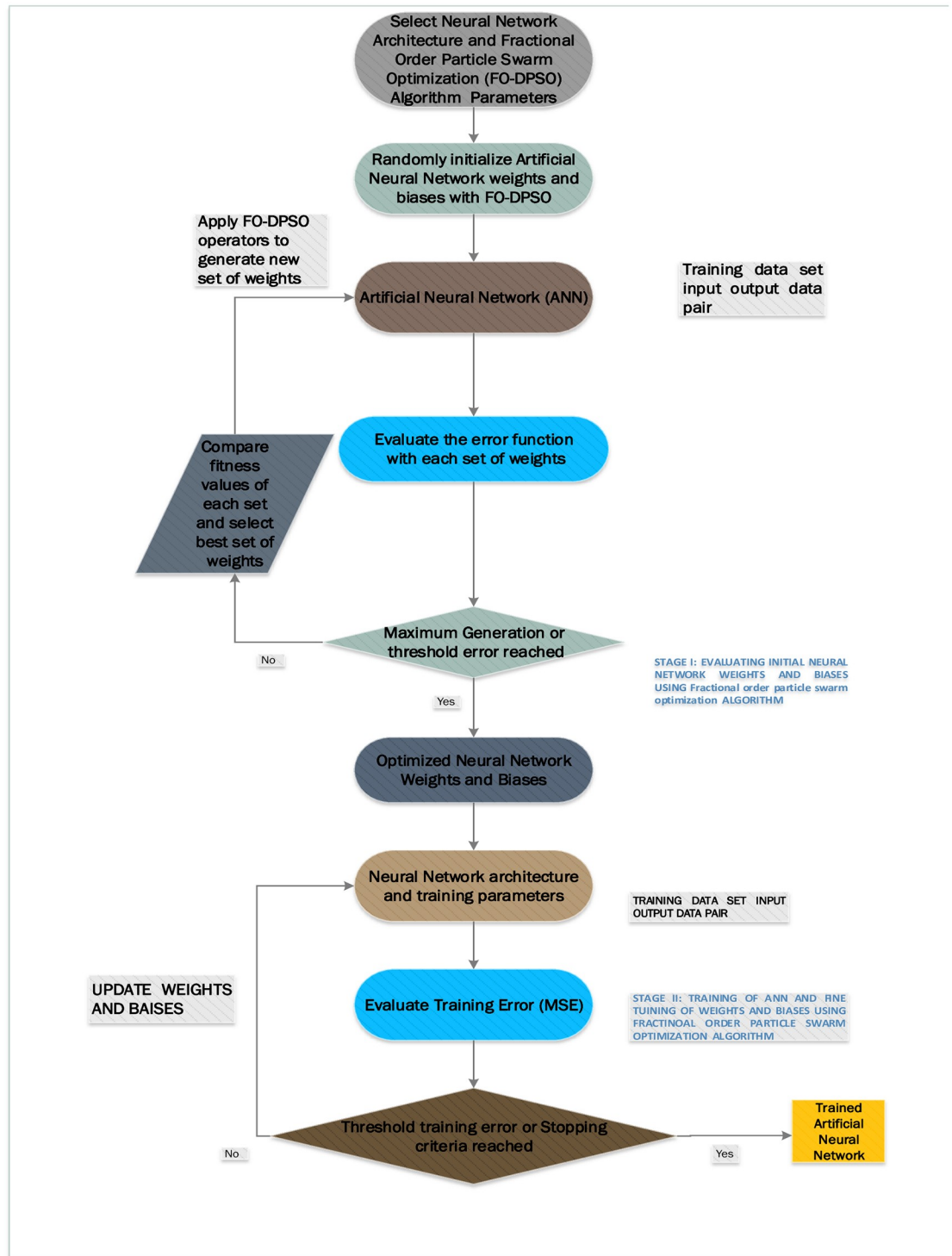


Fig 1. Graphical illustration of our soft computing procedure for doubly singular non-linear ODEs and Porous fin model.

<https://doi.org/10.1371/journal.pone.0235829.g001>

**Start of FO-DPSO**

**Step 1: Initialization:** Randomly generate the initial swarm. Assign values to the parameters of "FO-DPSO"

**Step 2: Fitness Evaluation:** Scrutinize the "fitness value" of each particle using equation (6)

**Step 3: Ranking:** Rank each particle of the minimum values of the "fitness function".

**Step 4: Stopping Criteria:** Stop if

- Level of "fitness" achieved
- Selected "flights/cycles" executed

If meet the "stopping" criteria, then move to Step 5

**Step 5: Renewal:** Call the "position" and "velocity", using equations (12) and (13).

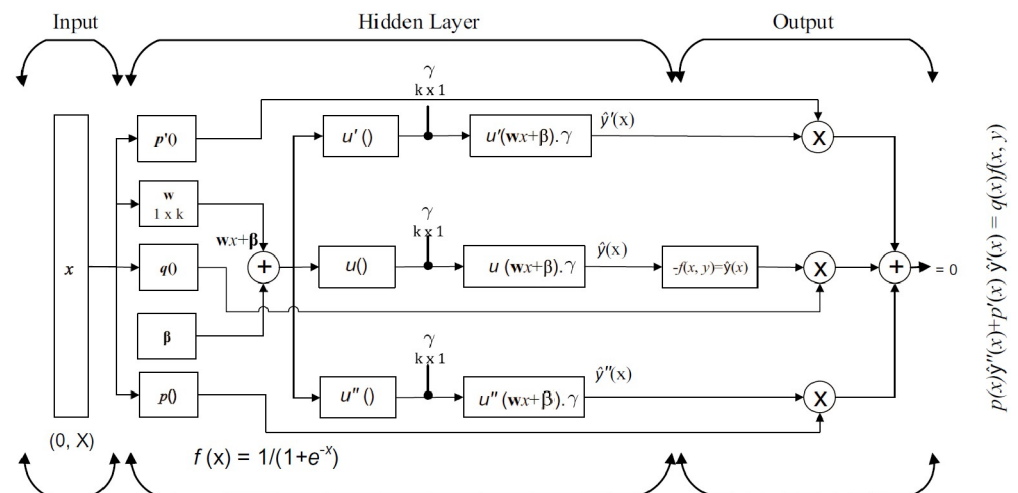
**Step 6: Improvement:** Repeat the steps 2 to 6 up to the whole flights are achieved.

**Step 7: Storage:** Store "fitness values", which are best achieved values and signify as "best global particle"

**End FO-DPSO**

**Fig 2. Pseudo-code of our soft computing technique.**

<https://doi.org/10.1371/journal.pone.0235829.g002>



**Fig 3. Neural network architecture.**

<https://doi.org/10.1371/journal.pone.0235829.g003>

**2.1 ANN based approximation**

Most of the real-world problems are mathematically modeled in the form of differential equations containing the derivative of integer and/ or fractional order. ANNs are frequently used to suggest approximate solutions for such problems [36–41].

In [42] the author presented a generalized method of neural networks to tackle both ODE's as well as PDE's. This method is based on approximation of a function and the capability of feedforward artificial neural network is to construct a solution of the differential equation which is differentiable and in a closed analytic form. Evolutionary optimization algorithm has been applied to train the weights and biases of the network, which results in minimum mean squared error and turn a good approximate solution of the problem.

The mathematical form of an approximate solution for BVPs with doubly singularities is suggested by feed-forward ANN as given in Eq (4),

$$\hat{y}(x) = \sum_{i=1}^k \gamma_i u(w_i x + \beta_i), \tag{4}$$

$$\frac{d^n \hat{y}(x)}{dx^n} = \sum_{i=1}^k \gamma_i \frac{d^n}{dx^n} u(w_i x + \beta_i), \tag{5}$$

$\hat{y}(x)$  represents the approximate solution,  $x$  is the independent variable,  $\gamma_i$ ,  $\beta_i$  and  $w_i$  are the unknown weights, and Eq (5) expresses the  $n^{th}$  derivative of this approximate solution. A detailed discussion of neural networks is given in [36, 37, 41]. An activation function  $u(x) = \frac{1}{1+e^{-x}}$  also known as Log-sigmoid mapping function is used to train the unknown weights. We substitute the approximate solutions in Eqs (1)–(3) to design an ANN-based solution for doubly singular BVPs. We show a detailed architecture of interaction among input-output and hidden stages of ANN in Fig 3. The objective function includes the minimization of mean squared errors in the approximate solutions.

$$\text{Minimize } e = e_1 + e_2, \tag{6}$$

where  $e_1, e_2$  are associated mean squared errors in ODE and boundary conditions, respectively.

$$e_1 = \frac{1}{N} \sum_{m=1}^N ((p_m \hat{y}'_m)' - q_m f(x_m, \hat{y}_m))^2, \tag{7}$$

where  $x \in (0, 1)$ , step size  $h = 0.05$ ,  $N = \frac{1}{h}$ ,  $p_m = p(x_m)$ ,  $q_m = q(x_m)$ , and  $x_m = mh$ .

Error due to boundary conditions is represented in Eq (8)

$$e_2 = \frac{1}{2} ((\hat{y}_0 - a_1)^2 + (\hat{y}_N - c_1)^2). \tag{8}$$

If the boundary conditions are mixed as discussed in Eq (3), then  $e_2$  can be approximated as,

$$e_2 = \frac{1}{2} ((\hat{y}'_0)^2 + (a \hat{y}_N + b \hat{y}'_N - c_1)^2). \tag{9}$$

### 2.2 Search method of FO-DPSO

According to published findings, fractional calculus (FC) has received much interest to adapt it in the interpretation and solution of engineering challenges [43–45], applied mathematics, mechanical/dynamics [46, 47]. Grunwald-Letnikov defined a fractional derivative that contains fractional coefficients  $\alpha \in R$ , a real number, by adjusting an unknown function  $x(t)$  as in Eq (10),

$$D^\alpha [x(t)] = \lim_{h \rightarrow 0} \left[ \frac{1}{h} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha + 1) x(t - kh)}{\Gamma(k + 1) \Gamma(\alpha - k + 1)} \right], \tag{10}$$

where the symbol  $\Gamma$  represent gamma function.

It is further elaborated that the series is characterized by bounded terms in Eq (10), if the derivative is of integer order. If  $\alpha$  is fractional, infinite terms represent the result. It is therefore important to note that ordinary derivatives are operators which are local / instantaneous,

whereas fractional operators represent a memory of past variations. With time, the memory of past instances declines. The Eq (11) determines a derivative for discrete instances. [48–52],

$$D^\alpha[x(t)] = \frac{1}{T^\alpha} \left[ \sum_{k=0}^r \frac{(-1)^k \Gamma(\alpha + 1) x(t - kh)}{\Gamma(k + 1) \Gamma(\alpha - k + 1)} \right], \tag{11}$$

The term  $T$  refers to the time intervals of events and  $r$  is number of truncated terms. Because of their memory retention properties, methods found in fractional calculus are useful in irretrievable and disorganised systems. Taking into account swarms' chaotic behavior in the Darwinian Particle swarms optimization algorithm, fractional calculus tools are appropriate to keep track of swarms' past movements.

Taking into account the inertial weight in FO-DPSO  $w = 1$ ,  $T$  as 1 and the research performed in [35, 53, 54], we have the following expression:

$$D^\alpha[v_{t+1}^n] = \rho_1 r_1 (\check{g}_t^n - x_t^n) + \rho_2 r_2 (\check{x}_t^n - x_t^n) + \rho_3 r_3 (\check{n}_t^n - x_t^n). \tag{12}$$

The empirical results of the algorithm are identical for  $r \geq 4$ . The computational complexity also increases almost linearly, and therefore takes up the memory of  $O(r)$ . Hence, it truncates the fifth term and onward for faster convergence. Thus  $r$ 's value is kept as 4. The inclusion of these four differential derivative terms means that the velocity term in FO-DPSO is as in Eq (13),

$$v_{t+1}^n = \alpha v_t^n + \frac{1}{2} \alpha v_{t-1}^n + \frac{1}{6} \alpha (1 - \alpha) v_{t-2}^n + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) v_{t-3}^n + \rho_1 r_1 (\check{g}_t^n - x_t^n) + \rho_2 r_2 (\check{x}_t^n - x_t^n) + \rho_3 r_3 (\check{n}_t^n - x_t^n). \tag{13}$$

### 3 Test problems and empirical results

In this section, we present three doubly singular type of differential equations, and their nine case studies are considered here to test the efficiency of our new approach.

#### 3.1 Problem 1

This problem is a linear ODE with a doubly singularity with a polynomial forcing term. It is a boundary value problem with a non-homogenous ODE. Mathematically, it can be represented as in Eq (14) [2],

$$\begin{cases} y''(x) + \frac{1}{x} y'(x) + \mu y = f(x), \\ y(0) = 1, \quad y(1) = 3, \end{cases} \tag{14}$$

where the exact solution is given in Eq (15)

$$y(x) = x^3 + x + 1. \tag{15}$$

Below we consider three cases of the problem by taking  $\mu = -9, -1, 1$  and forcing term as  $f_1(x) = -9 - 9x^3 + \frac{1}{x}$ ,  $f_2(x) = -1 - x^3 + 8x + \frac{1}{x}$  and  $f_3(x) = 1 + x^3 + 10x + \frac{1}{x}$  respectively.

Case 1: From Eq (14) with  $\mu = -9$  and  $f(x) = f_1(x)$  we get Eq (16),

$$y''(x) + \frac{1}{x} y'(x) - 9y = -9x^3 - 9 + \frac{1}{x}. \tag{16}$$

We give the error function which is used to measure the quality of the approximate solution in

Eq (17)

$$E = \frac{1}{N} \sum_{m=1}^N (x_m \hat{y}_m'' + \hat{y}_m' - 9x_m \hat{y}_m + 9x_m + 9x_m^4 - 1)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - 3)^2) \quad (17)$$

Case 2: From Eq (14) with  $\mu = -1$  and  $f(x) = f_2(x)$  we get Eq (18),

$$y''(x) + \frac{1}{x}y'(x) - y = -x^3 + 8x - 1 + \frac{1}{x}. \quad (18)$$

We give the error function which is used to measure the quality of the approximate solution in Eq (19)

$$E = \frac{1}{N} \sum_{m=1}^N (x_m \hat{y}_m'' + \hat{y}_m' - x_m \hat{y}_m + x_m + x_m^4 - 8x_m^2 - 1)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - 3)^2) \quad (19)$$

Case 3: From Eq (14) with  $\mu = 1$  and  $f(x) = f_3(x)$  we get Eq (20),

$$y''(x) + \frac{1}{x}y'(x) + y = x^3 + 10x + 1 + \frac{1}{x}. \quad (20)$$

We give the error function which is used to measure the quality of the approximate solution in Eq (21)

$$E = \frac{1}{N} \sum_{m=1}^N (x_m \hat{y}_m'' + \hat{y}_m' + x_m \hat{y}_m - x_m - x_m^4 - 10x_m^2 - 1)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - 3)^2) \quad (21)$$

The unknown decision weights in error functions (17), (19) and (21) are determined by using the novel ANN based FO-DPSO approach. We briefly present approximate solutions for cases 1, 2, and 3 in Eqs (22), (23) and (24). We show the best values of weights along with convergence plots and step sizes in Figs 4, 5, 6 for Problems 1, 2, and 3. After getting the best weights, we used their values in approximate solutions given in Eq (4). For reproduction of our results, we have presented the complete solutions in the Appendix section without rounding off errors.

$$\hat{y}_1(x) = \frac{-0.8218}{1 + e^{-(0.6157x+1.57378)}} + \dots + \frac{3.5514}{1 + e^{-(0.1380x+0.1581)}} \quad (22)$$

$$\hat{y}_2(x) = \frac{5.597}{1 + e^{-(2.3355x-3.0491)}} + \dots + \frac{-0.3914}{1 + e^{-(4.1093x+2.4261)}} \quad (23)$$

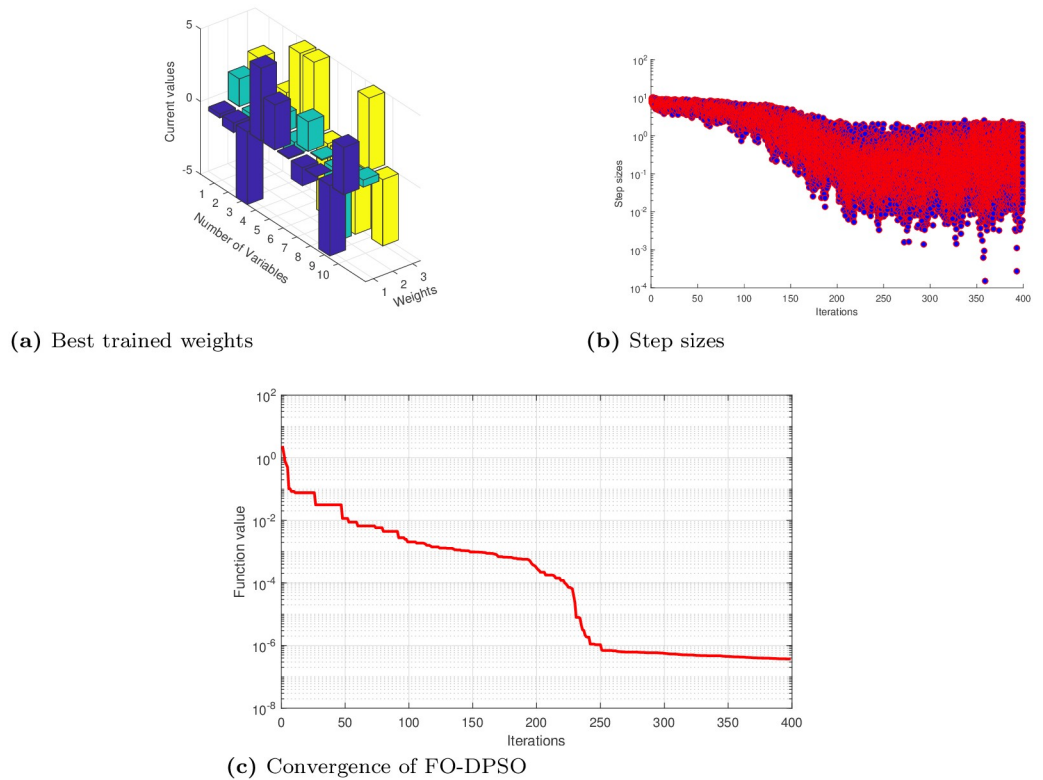
$$\hat{y}_3(x) = \frac{5.7670}{1 + e^{-(1.9282x+4.0689)}} + \dots + \frac{-2.8305}{1 + e^{-(0.1369x+0.3719)}} \quad (24)$$

Results got by FO-DPSO are compared with exact solutions, Genetic algorithm, its variant GA-SQP and are presented in Tables 1, 2, 3, 4, 5, 6, and 7 with step sizes  $h = 0.05$  and  $0.2$ , for problems 1, 2, 3 and 4. Input variable  $x$  is varied in interval  $[0, 1]$ . The Absolute Error (AE) is calculated to highlight the better performance of our approach. Mathematically, it can be expressed as in Eq (25):

$$AE = |y(x) - \hat{y}(x)|. \quad (25)$$

Values of AEs show better results in terms of accuracy of our approach. We give all values of





**Fig 4. Best weights obtained, convergence of error values and step sizes used to reach the best solution for problem 1 case 1, 2, 3 using feed-forward ANNs based on FO-DPSO algorithm.**

<https://doi.org/10.1371/journal.pone.0235829.g004>

AEs for step size  $h = 0.2$  in Tables 2, 4 and 6. A graphical illustration of AEs with  $h = 0.05$  is given in Fig 7.

FO-DPSO is better than GA-SQP and GA based approach as our approach is more accurate in solving problem 1. Results of AE show that ANN based FO-DPSO has produced values laying in ranges  $10^{-7}$  to  $10^{-12}$ ,  $10^{-8}$  to  $10^{-12}$  and  $10^{-8}$  to  $10^{-11}$  for problem 1, case 1, 2, and 3 respectively. We establish that ANN based on FO-DPSO is a successful technique for solving the problem under consideration.

### 3.2 Problem 2

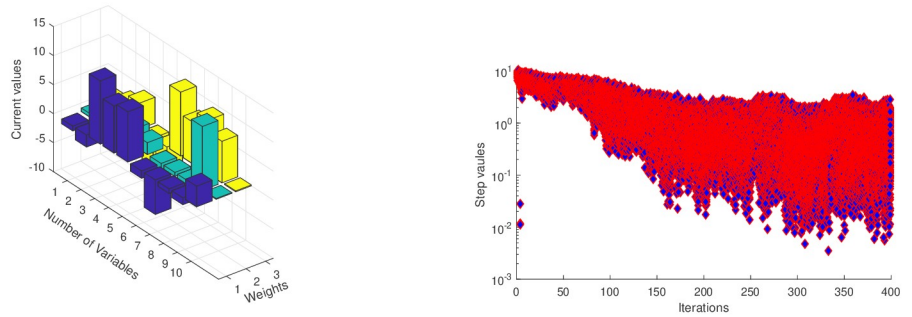
In this problem, we consider a doubly singular ODE with boundary values and variable coefficients. It is a homogenous differential equation of second order. Mathematically, this problem can be represented as in Eq (26) [2],

$$\begin{cases} (x^\mu y'(x))' = vx^{\mu+v-2}(vx^\nu + \mu + v - 1)y & 0 < x \leq 1 \quad \mu, v > 0 \\ y(0) = 1, \quad y(1) = e. \end{cases} \tag{26}$$

An exact solution for this problem is suggested in [2], Eq (27)

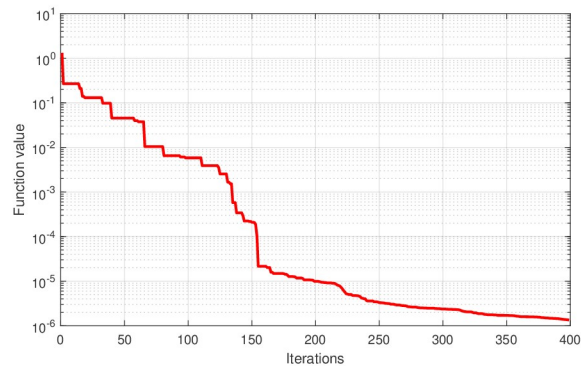
$$y = e^{x^\nu}. \tag{27}$$

We formulate three cases according to the variable coefficients  $\mu, v$ . We have tested the proposed method by solving these three cases.



(a) Best trained weights

(b) Step sizes



(c) Convergence of FO-DPSO

**Fig 5. Best weights obtained, convergence of error values and step sizes used to reach the best solution for problem 2 case 1, 2, 3 using feed-forward ANNs based on FO-DPSO algorithm.**

<https://doi.org/10.1371/journal.pone.0235829.g005>

Case 1: choosing  $\mu = 0.5$  and  $\nu = 1$  in problem (26), we get,

$$\sqrt{x}y''(x) + \frac{1}{2\sqrt{x}}y'(x) = \frac{1}{\sqrt{x}}\frac{2x+1}{2}y. \tag{28}$$

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (2x_m \hat{y}_m'' + \hat{y}_m' - 2x_m \hat{y}_m - \hat{y}_m)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - e)^2). \tag{29}$$

Case 2: choosing  $\mu = 0.75$  and  $\nu = 1$  in problem (26), we get,

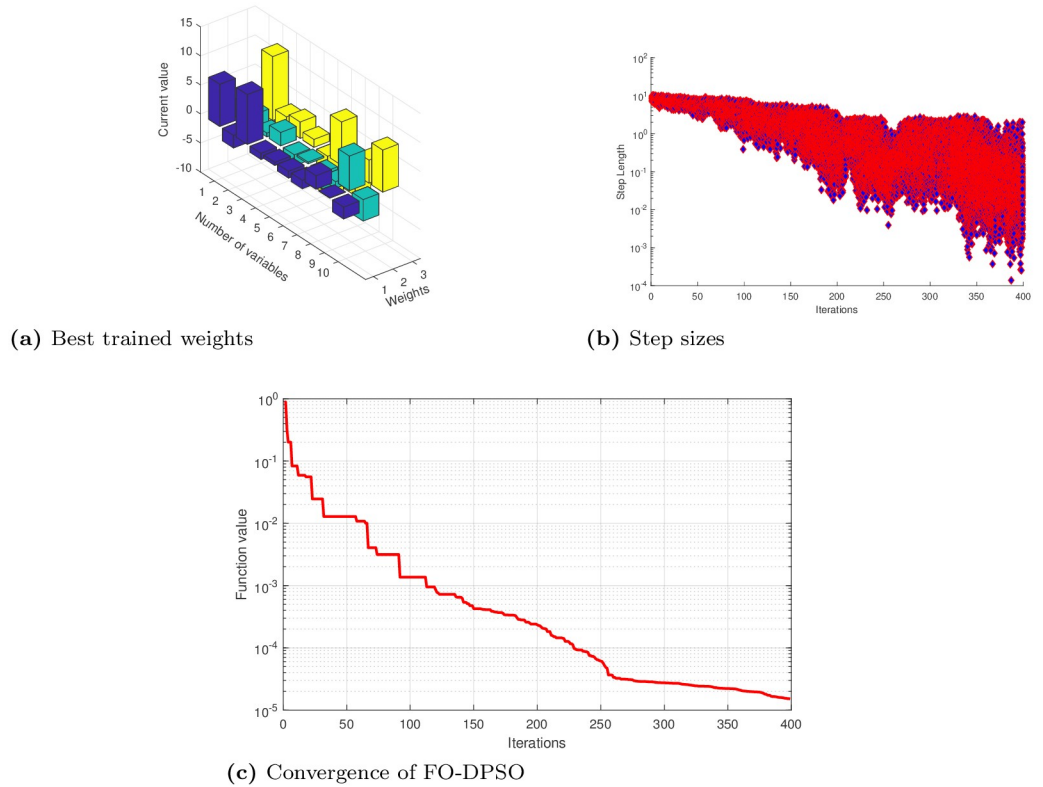
$$x^{3/4}y''(x) + \frac{3}{4x^{1/4}}y'(x) = \frac{1}{x^{1/4}}(x + \frac{3}{4})y. \tag{30}$$

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (4x_m \hat{y}_m'' + 3\hat{y}_m' - 4x_m \hat{y}_m - 3\hat{y}_m)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - e)^2). \tag{31}$$

Case 3: choosing  $\mu = 0.25$  and  $\nu = 1$  in problem (26), we get,

$$x^{1/4}y''(x) + \frac{1}{4x^{3/4}}y'(x) = \frac{1}{x^{3/4}}(x + \frac{1}{4})y. \tag{32}$$



**Fig 6. Best weights obtained, convergence of error values and step sizes used to reach the best solution for problem 3 case 1, 2, 3 using feed-forward ANNs based on FO-DPSO algorithm.**

<https://doi.org/10.1371/journal.pone.0235829.g006>

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (4x_m \hat{y}_m'' + \hat{y}_m' - 4x_m \hat{y}_m - \hat{y}_m)^2 + \frac{1}{2} ((\hat{y}_0 - 1)^2 + (\hat{y}_N - e)^2). \tag{33}$$

The unknown decision weights in error functions (29), (31) and (33) are determined by using the novel ANN based FO-DPSO approach. Values of AE show better results in terms of accuracy of the corresponding approach. We give all values of solutions and AEs (for  $h = 0.2$ ) in Tables 3 and 4 for problem 2. FO-DPSO is better than GA-SQP and GA based approach by solving problem 2 more accurately. Results of AE show that ANN-based FO-DPSO has produced, ranges in  $10^{-8}$  to  $10^{-11}$ ,  $10^{-8}$  to  $10^{-11}$  and  $10^{-9}$  to  $10^{-11}$  respectively. We establish it that ANN-based on FO-DPSO is a successful technique for solving the problem under consideration.

### 3.3 Problem 3

In this problem, we consider a nonlinear doubly singular ODE with boundary values and variable coefficients. It is a homogenous differential equation of second order. Mathematically, this problem can be represented as in Eq (34) [2],

$$\begin{cases} (x^\mu y'(x))' = \nu x^{\mu+\nu-2} e^y (e^y x^\nu - \mu - \nu + 1) & 0 < x \leq 1 \quad \mu, \nu > 0 \\ y(0) = \ln(\frac{\nu}{4}), \quad y(1) = \ln(\frac{\nu}{5}). \end{cases} \tag{34}$$

**Table 1. Empirical solutions for problem 1 (Case 1, 2, 3) achieved by FO-DPSO and GA.** Which are compared with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.05$ .

x	Case 1			Case 2		Case 3	
	Exact	FO-DPSO	GA	FO-DPSO	GA	FO-DPSO	GA
0	1	1.000031	1	1.000025	1	1	1
0.05	1.050125	1.050163	1.050124	1.050145	1.050125	1.050085	1.050122
0.1	1.101	1.101024	1.101	1.101008	1.101	1.100973	1.100999
0.15	1.153375	1.15338	1.153375	1.153373	1.153375	1.153365	1.153375
0.2	1.208	1.207991	1.208	1.207993	1.208	1.208002	1.208
0.25	1.265625	1.265613	1.265625	1.265617	1.265625	1.26563	1.265624
0.3	1.327	1.326993	1.327	1.326992	1.327	1.327002	1.326999
0.35	1.392875	1.392879	1.392875	1.392869	1.392875	1.39287	1.392875
0.4	1.464	1.464015	1.464	1.463995	1.464	1.463989	1.464
0.45	1.541125	1.541146	1.541125	1.54112	1.541125	1.54112	1.541125
0.5	1.625	1.625021	1.625	1.624993	1.625	1.62499	1.625001
0.55	1.716375	1.716391	1.716375	1.716365	1.716375	1.716369	1.716376
0.6	1.816	1.816008	1.816	1.815985	1.816	1.815998	1.816
0.65	1.924625	1.924625	1.924625	1.924607	1.924625	1.924626	1.924625
0.7	2.043	2.042996	2.043	2.042979	2.043	2.043	2.043
0.75	2.171875	2.171872	2.171875	2.171854	2.171875	2.171873	2.171876
0.8	2.312	2.312002	2.312	2.31198	2.312	2.311996	2.312001
0.85	2.464125	2.464133	2.464125	2.464105	2.464125	2.464121	2.464126
0.9	2.629	2.629013	2.629	2.62898	2.629	2.628997	2.629001
0.95	2.807375	2.807389	2.807375	2.807353	2.807375	2.807373	2.807376
1	3	3.000013	3	3	3	3	3.000001

<https://doi.org/10.1371/journal.pone.0235829.t001>

An exact solution for this problem is suggested in [2], Eq (35)

$$y = \ln \frac{v}{4 + x^v}. \tag{35}$$

We formulate three cases according to the variable coefficients  $\mu, v$ . We have tested the proposed method by solving these three cases,

Case 1: choosing  $\mu = 0.25$  and  $v = 1$  in problem (34), we get,

$$\begin{cases} x^{1/4}y''(x) + \frac{1}{4x^{3/4}}y'(x) = \frac{e^x}{x^{3/4}}(e^x - \frac{1}{4}) & 0 < x \leq 1 \quad \mu, v > 0 \\ y(0) = \ln(\frac{1}{4}), \quad y(1) = \ln(\frac{1}{5}). \end{cases} \tag{36}$$

**Table 2. Absolute errors in results for problem 1 (Case 1, 2, 3) achieved by FO-DPSO and GA-SQP.** Which are matched with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.2$ .

x	Case 1 (AE)		Case 2 (AE)		Case 3 (AE)	
	FO-DPSO	GA-SQP	FO-DPSO	GA-SQP	FO-DPSO	GA-SQP
0	9.98E-09	1.36E-07	4.19E-08	2.17E-07	4.69E-09	4.40E-07
0.2	6.25E-07	9.96E-09	8.94E-08	4.39E-08	3.71E-07	2.45E-07
0.4	1.43E-06	4.07E-08	1.17E-06	1.03E-07	1.04E-06	1.94E-08
0.6	1.19E-06	9.64E-10	1.50E-06	9.63E-08	9.23E-07	3.75E-07
0.8	3.30E-07	3.95E-08	5.96E-07	1.60E-07	2.53E-07	7.77E-07
1	1.34E-08	8.07E-08	2.26E-08	1.73E-07	1.85E-08	1.02E-06

<https://doi.org/10.1371/journal.pone.0235829.t002>

**Table 3. Empirical solutions for problem 2 (Case 1, 2, 3) achieved by FO-DPSO and GA.** Which are compared with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.05$ .

x	Case 1			Case 2		Case 3	
	Exact	FO-DPSO	GA	FO-DPSO	GA	FO-DPSO	GA
0	1	1.000001	1	1.000002	1.000014	1	1.000001
0.05	1.051271	1.051271	1.051245	1.051272	1.05129	1.051271	1.051273
0.1	1.105171	1.105171	1.10516	1.105172	1.105187	1.105171	1.105172
0.15	1.161834	1.161834	1.161832	1.161835	1.161844	1.161834	1.161835
0.2	1.221403	1.221403	1.221406	1.221403	1.221408	1.221403	1.221403
0.25	1.284025	1.284026	1.284031	1.284026	1.284027	1.284025	1.284026
0.3	1.349859	1.349859	1.349866	1.349859	1.349859	1.349859	1.349859
0.35	1.419068	1.419068	1.419074	1.419068	1.419068	1.419067	1.419068
0.4	1.491825	1.491825	1.49183	1.491825	1.491825	1.491825	1.491825
0.45	1.568312	1.568312	1.568317	1.568312	1.568314	1.568312	1.568312
0.5	1.648721	1.648721	1.648725	1.648721	1.648723	1.648721	1.648721
0.55	1.733253	1.733253	1.733257	1.733253	1.733254	1.733253	1.733253
0.6	1.822119	1.822119	1.822123	1.822118	1.822119	1.822119	1.822118
0.65	1.915541	1.915541	1.915547	1.91554	1.91554	1.915541	1.91554
0.7	2.013753	2.013753	2.013761	2.013752	2.01375	2.013753	2.013752
0.75	2.117	2.117	2.117011	2.117	2.116997	2.117	2.117
0.8	2.225541	2.225541	2.225555	2.22554	2.225537	2.225541	2.22554
0.85	2.339647	2.339647	2.339663	2.339646	2.339642	2.339647	2.339646
0.9	2.459603	2.459603	2.459621	2.459603	2.459598	2.459603	2.459602
0.95	2.58571	2.58571	2.585728	2.585709	2.585704	2.58571	2.585708
1	2.718282	2.718282	2.7183	2.718281	2.718276	2.718282	2.718281

<https://doi.org/10.1371/journal.pone.0235829.t003>

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (4x_m \hat{y}_m'' + \hat{y}_m' - 4x_m e^{\hat{y}_m} + e^{\hat{y}_m})^2 + \frac{1}{2} ((\hat{y}_0 - \ln \frac{1}{4})^2 + (\hat{y}_N - \ln \frac{1}{5})^2). \tag{37}$$

Case 2: choosing  $\mu = 0.5$  and  $\nu = 1$  in problem (34), we get,

$$\begin{cases} x^{1/2} y''(x) + \frac{1}{2x^{1/2}} y'(x) = \frac{e^y}{x^{1/2}} (e^y x - \frac{1}{2}) & 0 < x \leq 1 \quad \mu, \nu > 0 \\ y(0) = \ln(\frac{1}{4}), \quad y(1) = \ln(\frac{1}{5}). \end{cases} \tag{38}$$

**Table 4. Absolute errors in results for problem 2 (Case 1, 2, 3) achieved by FO-DPSO and GA.** Which are matched with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.2$ .

x	Case 1 (AE)		Case 2 (AE)		Case 3 (AE)	
	FO-DPSO	GA-SQP	FO-DPSO	GA-SQP	FO-DPSO	GA-SQP
0	6.45E-08	5.41E-07	2.48E-07	1.54E-06	8.66E-07	4.74E-07
0.2	9.19E-08	2.22E-07	2.67E-06	2.09E-07	3.23E-06	1.84E-07
0.4	1.05E-08	5.38E-08	8.41E-06	2.50E-08	1.88E-05	3.49E-08
0.6	2.22E-08	3.52E-08	5.33E-06	3.27E-07	2.65E-05	5.87E-08
0.8	5.91E-09	4.08E-08	9.07E-07	4.63E-07	1.17E-05	1.64E-07
1	2.77E-10	7.29E-08	2.94E-08	6.47E-07	1.05E-06	2.62E-07

<https://doi.org/10.1371/journal.pone.0235829.t004>

**Table 5. Empirical solutions for problem 3 (Case 1, 2, 3) achieved by FO-DPSO and GA.** Which are compared with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.05$ .

x	Case 1			Case 2		Case 3	
	Exact	FO-DPSO	GA	FO-DPSO	GA	FO-DPSO	GA
0	-1.38629	-1.386296	-1.38629	-1.386294	-1.38629	-1.386294	-1.38629
0.05	-1.39872	-1.398717	-1.39872	-1.398716	-1.39872	-1.398717	-1.39872
0.1	-1.41099	-1.410987	-1.41099	-1.410986	-1.41099	-1.410987	-1.41099
0.15	-1.42311	-1.423108	-1.42311	-1.423107	-1.42311	-1.423108	-1.42311
0.2	-1.43508	-1.435084	-1.43508	-1.435083	-1.43508	-1.435085	-1.43508
0.25	-1.44692	-1.446919	-1.44692	-1.446918	-1.44692	-1.446919	-1.44692
0.3	-1.45862	-1.458615	-1.45862	-1.458614	-1.45862	-1.458615	-1.45862
0.35	-1.47018	-1.470175	-1.47018	-1.470175	-1.47018	-1.470176	-1.47018
0.4	-1.4816	-1.481604	-1.4816	-1.481604	-1.4816	-1.481604	-1.4816
0.45	-1.4929	-1.492903	-1.4929	-1.492903	-1.4929	-1.492904	-1.4929
0.5	-1.50408	-1.504076	-1.50408	-1.504076	-1.50408	-1.504077	-1.50408
0.55	-1.51513	-1.515126	-1.51513	-1.515126	-1.51513	-1.515127	-1.51513
0.6	-1.52606	-1.526055	-1.52606	-1.526055	-1.52606	-1.526056	-1.52606
0.65	-1.53687	-1.536866	-1.53687	-1.536866	-1.53687	-1.536867	-1.53687
0.7	-1.54756	-1.547561	-1.54756	-1.547561	-1.54756	-1.547563	-1.54756
0.75	-1.55814	-1.558143	-1.55814	-1.558143	-1.55814	-1.558145	-1.55814
0.8	-1.56862	-1.568614	-1.56862	-1.568615	-1.56862	-1.568616	-1.56862
0.85	-1.57898	-1.578977	-1.57898	-1.578978	-1.57898	-1.578979	-1.57898
0.9	-1.58924	-1.589233	-1.58924	-1.589234	-1.58924	-1.589235	-1.58924
0.95	-1.59939	-1.599386	-1.59939	-1.599387	-1.59939	-1.599388	-1.59939
1	-1.60944	-1.609436	-1.60944	-1.609437	-1.60944	-1.609438	-1.60944

<https://doi.org/10.1371/journal.pone.0235829.t005>

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (2x_m \hat{y}_m'' + \hat{y}_m' - 2x_m e^{\hat{y}_m} + e^{\hat{y}_m})^2 + \frac{1}{2} ((\hat{y}_0 - \ln \frac{1}{4})^2 + (\hat{y}_N - \ln \frac{1}{5})^2). \tag{39}$$

Case 3: choosing  $\mu = 0.75$  and  $\nu = 1$  in problem (34), we get,

$$\begin{cases} x^{3/4} y''(x) + \frac{1}{2x^{1/4}} y'(x) = \frac{e^y}{x^{1/4}} (e^y x - \frac{3}{4}) & 0 < x \leq 1 \quad \mu, \nu > 0 \\ y(0) = \ln(\frac{1}{4}), \quad y(1) = \ln(\frac{1}{5}). \end{cases} \tag{40}$$

**Table 6. Absolute errors in results for problem 3 (Case 1, 2, 3) achieved by FO-DPSO and GA-SQP.** Which are matched with exact solutions for inputs  $x$  varying from 0 to 1 with a step size  $h = 0.2$ .

x	Case 1 (AE)		Case 2 (AE)		Case 3 (AE)	
	FO-DPSO	GA	FO-DPSO	GA	FO-DPSO	GA
0	1.39E-08	1.30E-08	7.68E-12	2.06E-07	8.55E-10	1.22E-09
0.2	6.01E-09	1.14E-09	5.51E-12	1.70E-08	5.98E-09	1.16E-09
0.4	2.79E-10	6.65E-09	4.90E-11	6.57E-08	6.20E-09	3.97E-09
0.6	1.16E-08	8.70E-09	3.55E-13	1.57E-07	2.43E-09	2.83E-09
0.8	1.44E-08	1.23E-08	7.72E-11	1.90E-07	9.90E-09	3.27E-09
1	1.65E-09	1.60E-08	1.97E-11	2.31E-07	1.49E-09	3.65E-09

<https://doi.org/10.1371/journal.pone.0235829.t006>

Table 7. Comparison of solution obtained for the problem 4 of porous fin designed model using FO-DPSO.

$\xi$	Analytical	FO-DPSO	GA-SQP	GA
0	0.700465898	0.701211994	0.701171286	0.701382837
0.1	0.703355803	0.704109143	0.704069922	0.704273632
0.2	0.712042101	0.712810621	0.712779543	0.71301169
0.3	0.72657335	0.727352198	0.727330423	0.727587757
0.4	0.747026504	0.747792755	0.747775149	0.74800451
0.5	0.773500952	0.774210653	0.774192242	0.774320572
0.6	0.806110215	0.806702499	0.806683296	0.806665864
0.7	0.844971293	0.845383557	0.845368094	0.845227794
0.8	0.890191724	0.890388584	0.890380987	0.890213813
0.9	0.941854371	0.941871225	0.941870115	0.941798879
1	1	0.99999949	0.999999331	1.000067362
MSE		6.50E-08	1.00E-07	1.00E-04

<https://doi.org/10.1371/journal.pone.0235829.t007>

The error function to judge the quality of the solutions is formulated as

$$E = \frac{1}{N} \sum_{m=1}^N (4x_m \hat{y}_m'' + 3\hat{y}_m' - 4x_m e^{\hat{y}_m} + 3e^{\hat{y}_m})^2 + \frac{1}{2} ((\hat{y}_0 - \ln \frac{1}{4})^2 + (\hat{y}_N - \ln \frac{1}{5})^2). \tag{41}$$

$$\hat{y}(x) = \frac{-1.7690}{1 + e^{-(-1.6186x-4.3759)}} + \frac{0.6479}{1 + e^{-(-0.3783x-0.1091)}} \tag{42}$$

$$\hat{y}(x) = \frac{-2.8314}{1 + e^{-(0.8127x+2.7334)}} + \frac{0.1693}{1 + e^{-(1.0845x+0.3734)}} \tag{43}$$

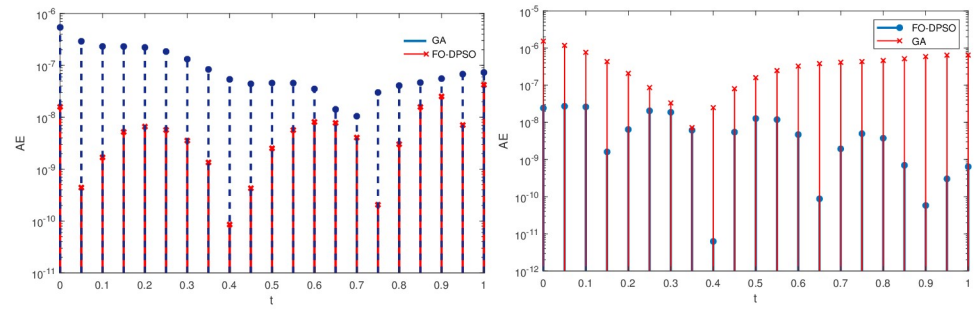
$$\hat{y}(x) = \frac{-0.2739}{1 + e^{-(1.9006x+2.8037)}} + \frac{3.2499}{1 + e^{-(0.4887x-4.5624)}} \tag{44}$$

The unknown decision weights in error functions (37), (39) and (41) are determined by using the novel ANN based FO-DPSO approach. We briefly present approximate solutions for all cases in Eqs (42), (43) and (44). Values of AEs show better performance in terms of accuracy of our approach. We give all solutions and AEs (for  $h = 0.2$ ) in Tables 5, and 6 for problem 3. FO-DPSO is better than GA and its variant GA-SQP by solving problem 3 more accurately. Results of AE show that ANN-based FO-DPSO has produced values laying in ranges  $10^{-11}$  to  $10^{-13}$ ,  $10^{-11}$  to  $10^{-16}$  and  $10^{-10}$  to  $10^{-15}$  respectively. We establish it that ANN-based on FO-DPSO is a successful technique for solving the problem under consideration.

### 4 Mathematical model of the porous fin

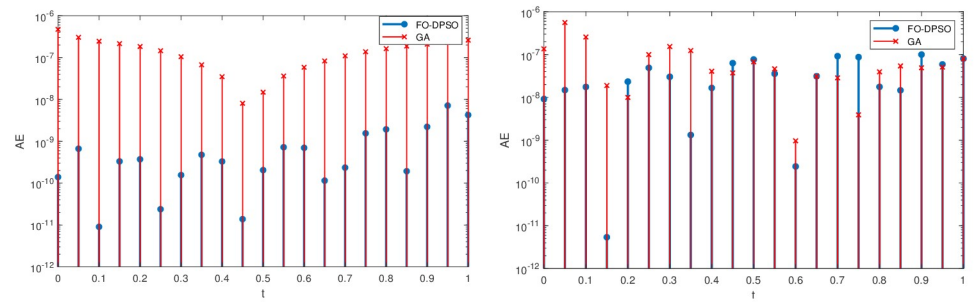
**Problem 4:** The schematic diagram of straight fin problem possessing the arbitrary cross-sectional area  $A_c$ , perimeter  $P$ , and length  $b$ , is presented in Fig 8, [55]. The fin is joined with the base surface having the temperature  $T_b$ , and extends into fluid having temperature  $T_a$ , and its tip is insulated. The energy balance equation is written as:

$$A_c \frac{d}{dx} \left[ k(T) \frac{dT}{dx} \right] - Ph(T - T_a) = 0. \tag{45}$$



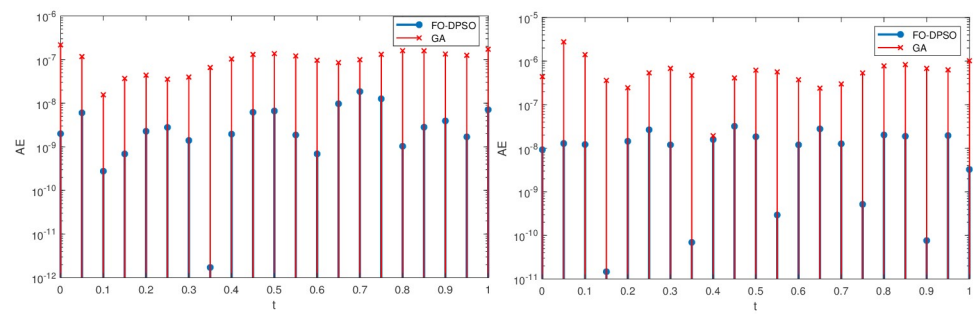
(a) Problem 1 Case 1

(b) Problem 1 Case 2



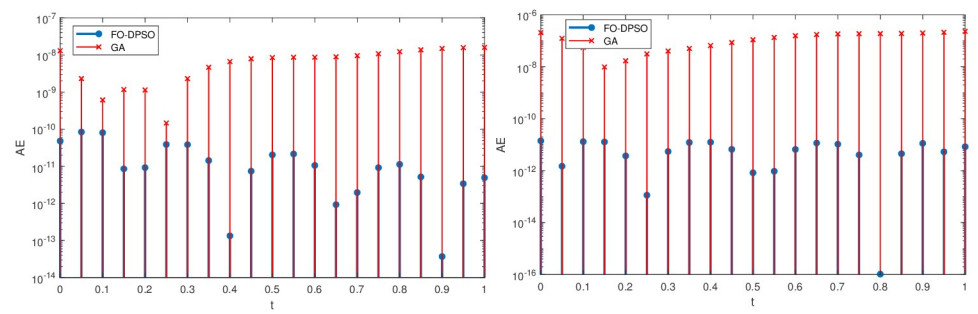
(c) Problem 1 Case 3

(d) Problem 2 Case 1



(e) Problem 2 Case 2

(f) Problem 2 Case 3



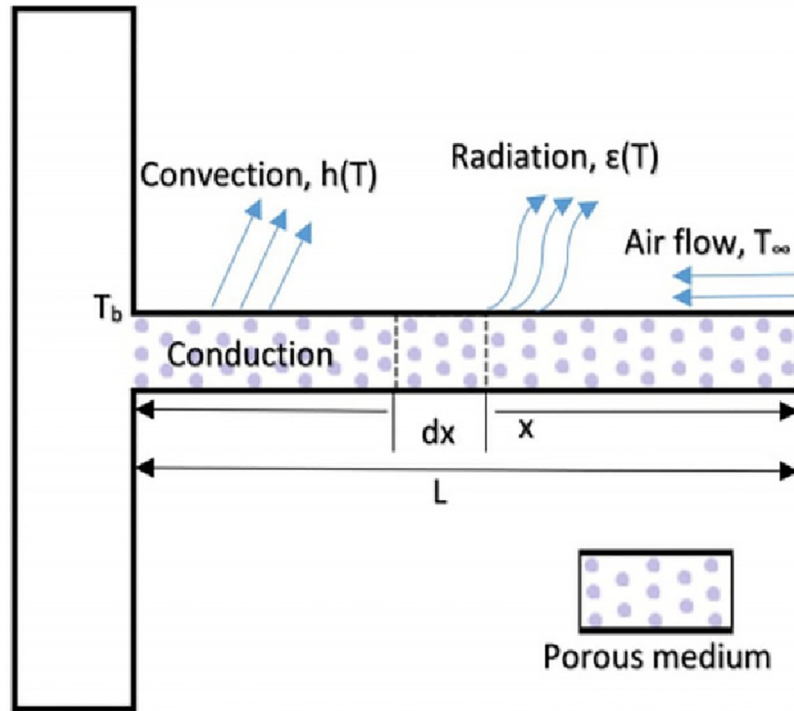
(g) Problem 3 Case 1

(h) Problem 3 Case 2

**Fig 7. Graphical illustration of absolute errors in best solutions, for problem 1, 2 and 3 (Case 1, 2, 3), obtained by FO-DPSO and GA.**

<https://doi.org/10.1371/journal.pone.0235829.g007>





**Fig 8. Design of a porous fin.**

<https://doi.org/10.1371/journal.pone.0235829.g008>

In the Eq (45),  $k(T)$  indicates the temperature-dependent thermal conductivity and  $h$  represents the coefficient of heat transfer. It is considered that the thermal conductivity for the fin material is expressed:

$$k(T) = k_b[1 + \lambda(T - T_b)]. \tag{46}$$

In the expression (46),  $k_b$  represents the thermal conductivity at the ambient fluid temperature of the fin, and  $\lambda$  is standing for the variation of the thermal conductivity. Using the non-dimensional variables:

$$\theta = \frac{T - T_a}{T_b - T_a}, \quad \xi = \frac{x}{b}, \quad \mu = \lambda(T_b - T_a), \quad \text{and} \quad \psi = \left(\frac{Phb^2}{k_a A_c}\right)^{1/2}, \tag{47}$$

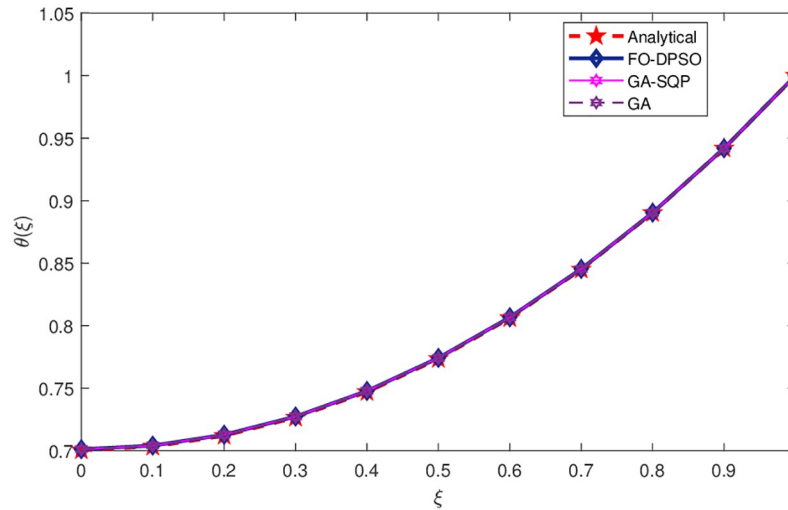
consequently the Eq (45) reduces into the following form:

$$\frac{d^2\theta}{d\xi^2} + \mu\theta \frac{d^2\theta}{d\xi^2} + \mu\left(\frac{d\theta}{d\xi}\right)^2 - \psi^2\theta, \quad 0 \leq \xi \leq 1, \tag{48}$$

with the boundary conditions

$$\left.\frac{d\theta}{d\xi}\right|_{\xi=0} = 0 \quad \text{and} \quad \theta|_{\xi=1} = 1. \tag{49}$$

We have solved a porous fin model using ANNs based FO-DPSO approach. The unknown weights are tuned by the FO-DPSO algorithm. Results obtained by FO-DPSO are compared with GA-SQP and GA and are given in Table 7. The graphical illustration of this model is given in Fig 8. Solutions obtained for this problem are plotted in Fig 9. Among the three



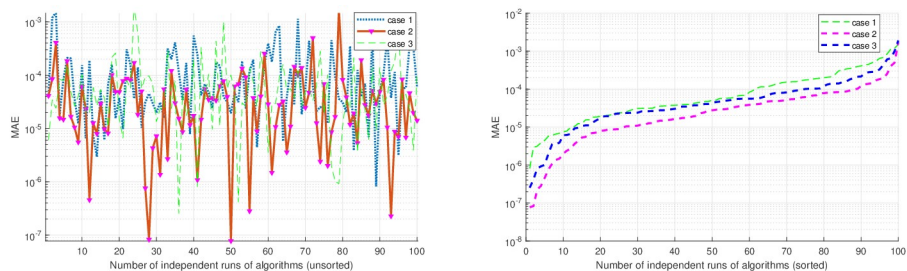
**Fig 9.** Solution obtained by our proposed method and other comparative algorithms for porous fin model.

<https://doi.org/10.1371/journal.pone.0235829.g009>

algorithms, FO-DPSO performed well and gave us minimum error as compared to the other algorithms, see Fig 9.

### 5 Statistical analysis

Performance indicators like global mean absolute error (GMAE), mean absolute error (MAE), and mean value of fitness ( $M_{fit}$ ) are used to assess the performance of ANNs based on FO-DPSO approach. We use these indicators on the data we have got through 100 independent simulations to determine the stability and robustness of our approach. We present MAE values in terms of sorted and unsorted form, see Figs 10, 11 and 12. Hence, the sorted results are presented in Figs 10b, 11b and 12b, while the unsorted errors in the solutions are given in Figs 10a, 11a and 12a, respectively. To further elaborate the difference between the errors obtained, and those reported in the literature, we have used the log scale plots for MAEs. From our graphical analysis, we get the minimum values of MAEs and better fitness for all problems. The performance of our approach is statistically analyzed in terms of the best minimum value, mean, and standard deviation (SD). This further validates our claim that our approach is better in convergence rate and has produced accurate results for all three BVPs with doubly singularities, and porous fin model. We present statistical results in terms of GMAE, Mean-time, Max

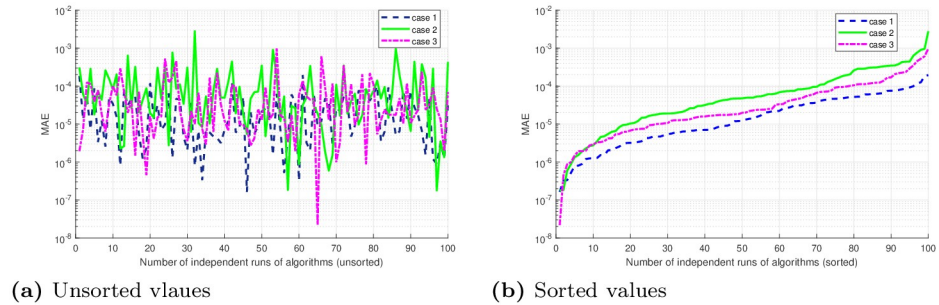


(a) Unsorted values

(b) Sorted values

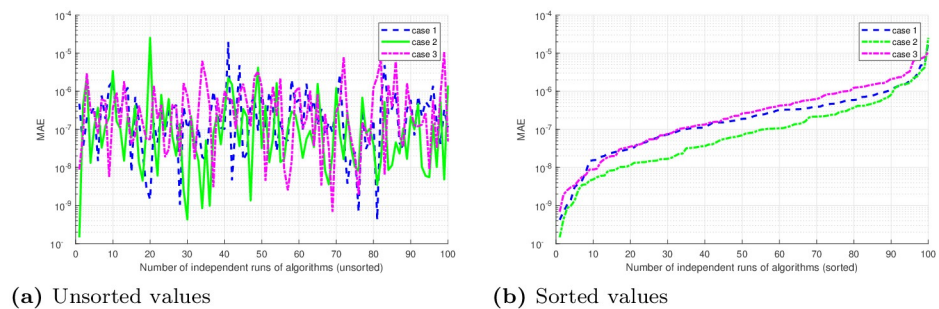
**Fig 10.** Graphical illustration of sorted absolute errors in solutions, for problem 1 (Case 1, 2, 3), obtained by FO-DPSO during 100 runs.

<https://doi.org/10.1371/journal.pone.0235829.g010>



**Fig 11. Graphical illustration of sorted absolute errors in solutions, for problem 2 (Case 1, 2, 3), obtained by FO-DPSO during 100 runs.**

<https://doi.org/10.1371/journal.pone.0235829.g011>



**Fig 12. Graphical illustration of sorted absolute errors in solutions, for problem 3 (Case 1, 2, 3), obtained by FO-DPSO during 100 runs.**

<https://doi.org/10.1371/journal.pone.0235829.g012>

iterations in Tables 8, 9 and 10. Our experimental outcome dictates that the ANNs based FO-DPSO approach has consistently produced better solutions to the non-linear ODEs and a real application problem.

To verify the stability and robustness of the proposed technique, we got better values of global performance indicators; global mean absolute error ( $G_{MAE}$ ) as in Eq (50) and mean of fitness values denoted as ( $M_{fit}$ ) as in Eq (51). All results in terms of these global performance indicators, see Tables 8, 9, and 10 revealed the fact that our approach is better than state-of-the-art approaches reported in the literature [33].

$$G_{MAE} = \frac{1}{R} \left( \sum_{r=1}^R \frac{1}{p} \left( \sum_{i=1}^p |y_i - \hat{y}_{i,r}| \right) \right), \tag{50}$$

**Table 8. Performance indicators based on proposed results for Problem 1.**

Type Parameters	Case 1	Case 2	Case 3
GMAE Values	9.40E-06	3.0273E-06	6.6613E-06
STD	5.6686E-06	2.7162E-06	5.3095E-06
Mfit VALUES	4.0931E-08	4.2931E-09	1.2824E-08
STD	3.2671E-08	4.6459E-09	9.6663E-09
Mean Time	27.8	30	28.4
Max Iteration	1000	1000	1000

<https://doi.org/10.1371/journal.pone.0235829.t008>

**Table 9. Performance indicators based on proposed results for Problem 2.**

Type Parameters	Case 1	Case 2	Case 3
GMAE Values	1.51E-06	4.164E-06	3.207E-06
STD	9.542E-07	3.328E-06	2.047E-06
Mfit VALUES	7.747E-09	8.493E-09	1.037E-09
STD	1.005E-08	9.361E-09	1.729E-09
Mean Time	40.9	27.7	30.7
Max Iteration	1000	1000	1000

<https://doi.org/10.1371/journal.pone.0235829.t009>

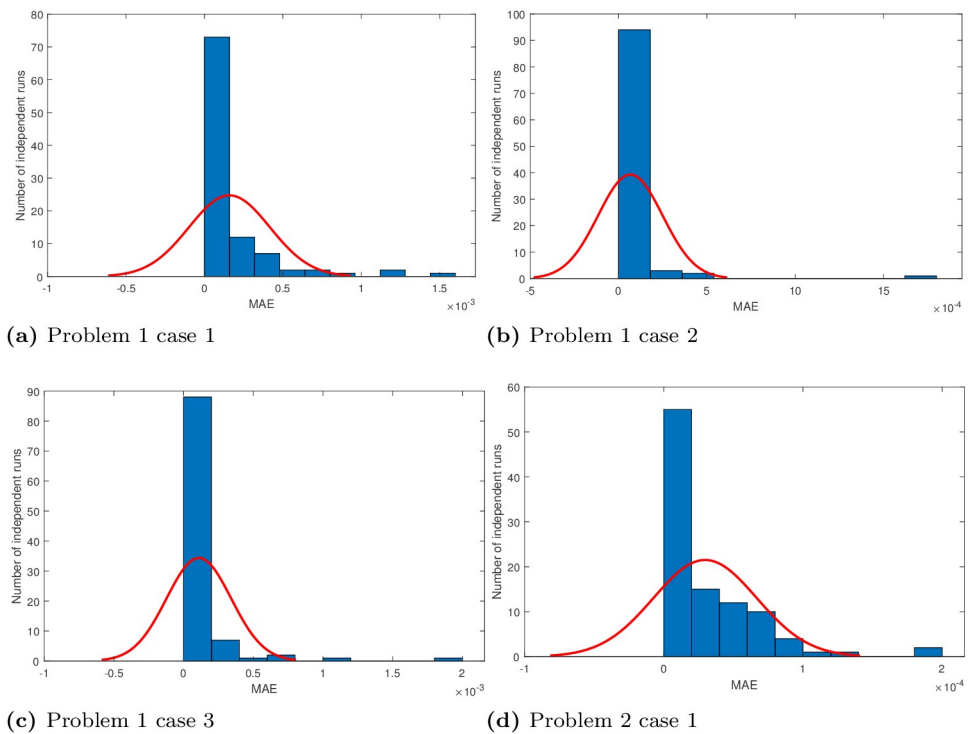
**Table 10. Performance indicators based on proposed results for Problem 3.**

Type Parameters	Case 1	Case 2	Case 3
GMAE Values	1.48E-08	5.4029E-09	1.2547E-08
STD	1.101E-08	3.7189E-09	1.0439E-08
Mfit VALUES	1.9996E-11	6.9589E-12	3.2591E-11
STD	2.4734E-11	4.8198E-12	6.6736E-11
Mean Time	29.5	29.9	28.9
Max Iteration	1000	1000	1000

<https://doi.org/10.1371/journal.pone.0235829.t010>

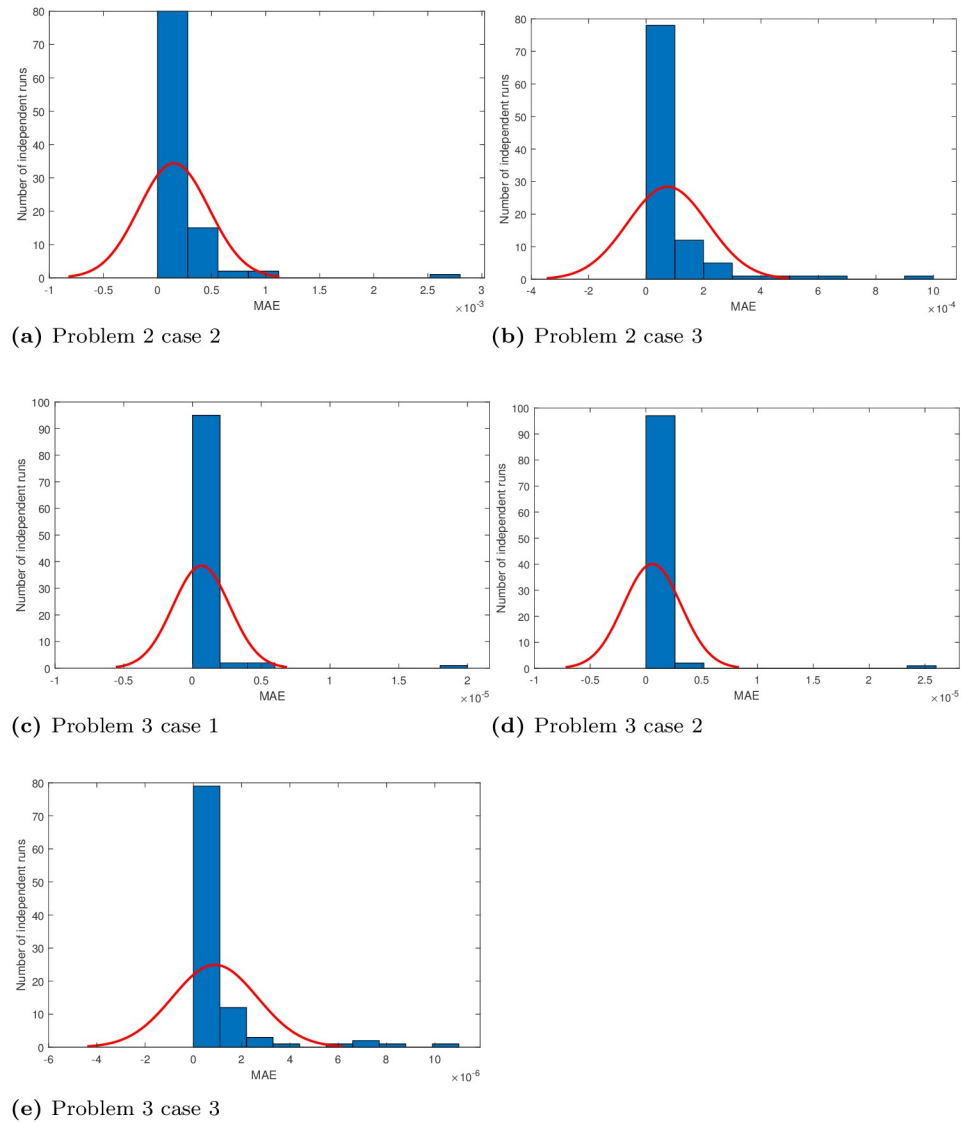
$$M_{Fit} = \frac{1}{R} \left( \sum_{r=1}^R E_r \right), \tag{51}$$

$$MeanTime = \frac{1}{R} \left( \sum_{r=1}^R T_r \right), \tag{52}$$



**Fig 13. Normal plots of MAE obtained by FO-DPSO during 100 runs.**

<https://doi.org/10.1371/journal.pone.0235829.g013>



**Fig 14. Normal plots of MAE obtained by FO-DPSO during 100 runs.**

<https://doi.org/10.1371/journal.pone.0235829.g014>

where the number of inputs and the number of simulations is denoted by  $P$ , and  $R$ , respectively, and  $y_i$  denote the corresponding exact solutions and  $\hat{y}_{i,r}$  the corresponding best approximate values obtained with our approach.  $E_r$  represents the best objective value achieved during the  $r^{th}$  simulation. In our experiments, input variable varies from 0 to 1 with  $h = 0.05$ , and  $0.2$  as step size. Thus, the total grid points were 20, and 6. We repeated our simulations for 100 times. Moreover, mean absolute errors in all solutions are negligible, and histograms with normal distribution fit for mean absolute errors (MAEs) in solutions of problems 1, 2, 3 are given in Figs 13 and 14.

## 6 Conclusion

In this paper, we present a new soft computing approach that combines artificial neural networks with a fractional-order particle swarm optimization (FO-DPSO) algorithm. We conclude this research by following key findings.

- The proposed method to exploit ANNs based FO-DPSO accurately solves variants of non-linear, doubly singular ordinary differential equations according to the computational evaluations.
- We calculate absolute errors in our results using the exact reference solutions, and the errors. Experimental results show that our designed scheme is more accurate compared to the state-of-the-art algorithms GA, and its latest variant GA-SQP.
- We have considered three hard problems with nine cases. A real application is also considered. We have analyzed a mathematical model of porous fins, and temperature profiles are studied for this model.
- We have analyzed several actual application problems. The Values of performance indicators,  $G_{MAE}$  and  $M_{fit}$  dictate that our approach gives results with lower errors compared to other algorithms.
- Frequency graphs of 100 experiments in terms of MAD are presented with normal distribution fittings. These graphs have proved that our approach is reliable and stable in terms of success rate.
- The solutions to the problems are in the appendix to help the reader reproduce the results presented in this paper.
- The proposed approach provides a more accurate solution to differential equations with multiple singularities and systems of such equations. Problems arising in thermodynamics, electromagnetic, and nanotechnology can be handled by the proposed method by changing the activation function of the artificial neural network.

## Supporting information

**S1 Appendix.**  
(PDF)

## Author Contributions

**Conceptualization:** Waseem Waseem.

**Formal analysis:** Waseem Waseem.

**Funding acquisition:** Poom Kumam.

**Methodology:** Waseem Waseem.

**Project administration:** Muhammad Sulaiman.

**Resources:** Muhammad Sulaiman, Muhammad Asif Zahoor Raja, Saeed Islam.

**Software:** Muhammad Sulaiman.

**Supervision:** Muhammad Sulaiman.

**Validation:** Waseem Waseem, Muhammad Sulaiman, Muhammad Asif Zahoor Raja.

**Visualization:** Waseem Waseem.

**Writing – original draft:** Waseem Waseem.

**Writing – review & editing:** Muhammad Sulaiman, Poom Kumam, Muhammad Shoaib.

## References

1. Singh R, Kumar J. An efficient numerical technique for the solution of nonlinear singular boundary value problems. *Computer Physics Communications*. 2014; 185(4):1282–1289. <https://doi.org/10.1016/j.cpc.2014.01.002>
2. Singh R, Kumar J. The Adomian decomposition method with Green's function for solving nonlinear singular boundary value problems. *Journal of Applied Mathematics and Computing*. 2014; 44(1-2):397–416. <https://doi.org/10.1007/s12190-013-0699-4>
3. Singh R, Kumar J, Nelakanti G. Approximate series solution of singular boundary value problems with derivative dependence using Green's function technique. *Computational and Applied Mathematics*. 2014; 33(2):451–467. <https://doi.org/10.1007/s40314-013-0074-y>
4. Singh R, Kumar J, Nelakanti G. New approach for solving a class of doubly singular two-point boundary value problems using Adomian decomposition method. *Advances in Numerical Analysis*. 2012; 2012. <https://doi.org/10.1155/2012/541083>
5. Verma AK, Singh M. Singular nonlinear three point BVPs arising in thermal explosion in a cylindrical reactor. *Journal of Mathematical Chemistry*. 2015; 53(2):670–684. <https://doi.org/10.1007/s10910-014-0447-5>
6. Roul P. An improved iterative technique for solving nonlinear doubly singular two-point boundary value problems. *The European Physical Journal Plus*. 2016; 131(6):209. <https://doi.org/10.1140/epjp/i2016-16209-1>
7. Roul P, Warbhe U. A novel numerical approach and its convergence for numerical solution of nonlinear doubly singular boundary value problems. *Journal of Computational and Applied Mathematics*. 2016; 296:661–676. <https://doi.org/10.1016/j.cam.2015.10.020>
8. Pandey RK, Gupta G. A note on fourth order method for doubly singular boundary value problems. *Advances in Numerical Analysis*. 2012; 2012. <https://doi.org/10.1155/2012/349618>
9. Pandey R, Verma AK. On solvability of derivative dependent doubly singular boundary value problems. *Journal of Applied Mathematics and Computing*. 2010; 33(1-2):489–511. <https://doi.org/10.1007/s12190-009-0299-5>
10. Singh R, Wazwaz AM, Kumar J. An efficient semi-numerical technique for solving nonlinear singular boundary value problems arising in various physical models. *International Journal of Computer Mathematics*. 2016; 93(8):1330–1346. <https://doi.org/10.1080/00207160.2015.1045888>
11. Malek S. On the summability of formal solutions for doubly singular nonlinear partial differential equations. *Journal of dynamical and control systems*. 2012; 18(1):45–82. <https://doi.org/10.1007/s10883-012-9134-7>
12. Canalis-Durand M, Mozo-Fernández J, Schäfke R. Monomial summability and doubly singular differential equations. *Journal of Differential Equations*. 2007; 233(2):485–511. <https://doi.org/10.1016/j.jde.2006.11.005>
13. Mall S, Chakraverty S. Application of Legendre Neural Network for solving ordinary differential equations. *Applied Soft Computing*. 2016; 43:347–356. <https://doi.org/10.1016/j.asoc.2015.10.069>
14. Mall S, Chakraverty S. Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev Neural Network method. *Neurocomputing*. 2015; 149:975–982. <https://doi.org/10.1016/j.neucom.2014.07.036>
15. Mall S, Chakraverty S. Chebyshev neural network based model for solving Lane–Emden type equations. *Applied Mathematics and Computation*. 2014; 247:100–114. <https://doi.org/10.1016/j.amc.2014.08.085>
16. Chakraverty S, Mall S. Regression-based weight generation algorithm in neural network for solution of initial and boundary value problems. *Neural Computing and Applications*. 2014; 25(3-4):585–594. <https://doi.org/10.1007/s00521-013-1526-4>
17. Raja MAZ, Samar R, Alaidarous ES, Shivanian E. Bio-inspired computing platform for reliable solution of Bratu-type equations arising in the modeling of electrically conducting solids. *Applied Mathematical Modelling*. 2016; 40(11-12):5964–5977. <https://doi.org/10.1016/j.apm.2016.01.034>
18. Masood Z, Majeed K, Samar R, Raja MAZ. Design of Mexican Hat Wavelet neural networks for solving Bratu type nonlinear systems. *Neurocomputing*. 2017; 221:1–14. <https://doi.org/10.1016/j.neucom.2016.08.079>
19. Raja MAZ. Solution of the one-dimensional Bratu equation arising in the fuel ignition model using ANN optimised with PSO and SQP. *Connection Science*. 2014; 26(3):195–214. <https://doi.org/10.1080/09540091.2014.907555>
20. Ahmad I, Raja MAZ, Bilal M, Ashraf F. Neural network methods to solve the Lane–Emden type equations arising in thermodynamic studies of the spherical gas cloud model. *Neural Computing and Applications*. 2017; 28(1):929–944. <https://doi.org/10.1007/s00521-016-2400-y>

21. Raja MAZ, Ahmad I, Khan I, Syam MI, Wazwaz AM. Neuro-heuristic computational intelligence for solving nonlinear pantograph systems. *Frontiers of Information Technology & Electronic Engineering*. 2017; 18(4):464–484. <https://doi.org/10.1631/FITEE.1500393>
22. Raja MAZ, Zameer A, Khan AU, Wazwaz AM. A new numerical approach to solve Thomas–Fermi model of an atom using bio-inspired heuristics integrated with sequential quadratic programming. *SpringerPlus*. 2016; 5(1):1400. <https://doi.org/10.1186/s40064-016-3093-5> PMID: 27610319
23. Effati S, Pakdaman M. Artificial neural network approach for solving fuzzy differential equations. *Information Sciences*. 2010; 180(8):1434–1457. <https://doi.org/10.1016/j.ins.2009.12.016>
24. Effati S, Skandari MHN. Optimal control approach for solving linear Volterra integral equations. *International Journal of Intelligent Systems and Applications*. 2012; 4(4):40. <https://doi.org/10.5815/ijisa.2012.04.06>
25. Jafarian A, Measoomy S, Abbasbandy S. Artificial neural networks based modeling for solving Volterra integral equations system. *Applied Soft Computing*. 2015; 27:391–398. <https://doi.org/10.1016/j.asoc.2014.10.036>
26. Raja MAZ, Farooq U, Chaudhary NI, Wazwaz AM. Stochastic numerical solver for nanofluidic problems containing multi-walled carbon nanotubes. *Applied Soft Computing*. 2016; 38:561–586. <https://doi.org/10.1016/j.asoc.2015.10.015>
27. Effati S, Buzhabadi R. A neural network approach for solving Fredholm integral equations of the second kind. *Neural Computing and Applications*. 2012; 21(5):843–852. <https://doi.org/10.1007/s00521-010-0489-y>
28. Raja MAZ, Khan JA, Chaudhary NI, Shivanian E. Reliable numerical treatment of nonlinear singular Flierl–Petviashvili equations for unbounded domain using ANN, GAs, and SQP. *Applied Soft Computing*. 2016; 38:617–636. <https://doi.org/10.1016/j.asoc.2015.10.017>
29. Sabouri J, Effati S, Pakdaman M. A neural network approach for solving a class of fractional optimal control problems. *Neural Processing Letters*. 2017; 45(1):59–74. <https://doi.org/10.1007/s11063-016-9510-5>
30. Effati S, Mansoori A, Eshaghnezhad M. An efficient projection neural network for solving bilinear programming problems. *Neurocomputing*. 2015; 168:1188–1197. <https://doi.org/10.1016/j.neucom.2015.05.003>
31. Kumar M, Yadav N. Numerical solution of Bratu's problem using multilayer perceptron neural network method. *National Academy Science Letters*. 2015; 38(5):425–428. <https://doi.org/10.1007/s40009-015-0359-3>
32. Khan JA, Raja MAZ, Rashidi MM, Syam MI, Wazwaz AM. Nature-inspired computing approach for solving non-linear singular Emden–Fowler problem arising in electromagnetic theory. *Connection Science*. 2015; 27(4):377–396. <https://doi.org/10.1080/09540091.2015.1092499>
33. Raja MAZ, Mehmood J, Sabir Z, Nasab AK, Manzar MA. Numerical solution of doubly singular nonlinear systems using neural networks-based integrated intelligent computing. *Neural Computing and Applications*. 2019; 31(3):793–812. <https://doi.org/10.1007/s00521-017-3110-9>
34. Couceiro MS, Rocha RP, Ferreira NF, Machado JT. Introducing the fractional-order Darwinian PSO. *Signal, Image and Video Processing*. 2012; 6(3):343–350. <https://doi.org/10.1007/s11760-012-0316-2>
35. Couceiro M, Ghamisi P. Fractional-order Darwinian PSO. In: *Fractional order darwinian particle swarm optimization*. Springer; 2016. p. 11–20.
36. Yadav N, Yadav A, Kumar M, Kim JH. An efficient algorithm based on artificial neural networks and particle swarm optimization for solution of nonlinear Troesch's problem. *Neural Computing and Applications*. 2017; 28(1):171–178. <https://doi.org/10.1007/s00521-015-2046-1>
37. Raja MAZ. Stochastic numerical treatment for solving Troesch's problem. *Information Sciences*. 2014; 279:860–873. <https://doi.org/10.1016/j.ins.2014.04.036>
38. Raja MAZ, Khan MAR, Mahmood T, Farooq U, Chaudhary NI. Design of bio-inspired computing technique for nanofluidics based on nonlinear Jeffery–Hamel flow equations. *Canadian Journal of Physics*. 2016; 94(5):474–489. <https://doi.org/10.1139/cjp-2015-0440>
39. Raja MAZ, Khan JA, Haroon T. Stochastic numerical treatment for thin film flow of third grade fluid using unsupervised neural networks. *Journal of the Taiwan Institute of Chemical Engineers*. 2015; 48:26–39. <https://doi.org/10.1016/j.jtice.2014.10.018>
40. Raja MAZ, Manzar MA, Samar R. An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP. *Applied Mathematical Modelling*. 2015; 39(10–11):3075–3093. <https://doi.org/10.1016/j.apm.2014.11.024>
41. Raja MAZ, Samar R, Manzar MA, Shah SM. Design of unsupervised fractional neural network model optimized with interior point algorithm for solving Bagley–Torvik equation. *Mathematics and Computers in Simulation*. 2017; 132:139–158. <https://doi.org/10.1016/j.matcom.2016.08.002>



42. Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*. 1998; 9(5):987–1000. <https://doi.org/10.1109/72.712178> PMID: 18255782
43. Machado JT, Sabatier J. *Advances in fractional calculus: theoretical developments and applications in physics and engineering*. Springer Verlag; 2007.
44. Ortigueira M, Tenreiro Machado J. Special issue on fractional signal processing. *Signal Process*. 2003; 83(11):2285–2286.
45. Tenreiro Machado J, Silva MF, Barbosa RS, Jesus IS, Reis CM, Marcos MG, et al. Some applications of fractional calculus in engineering. *Mathematical Problems in Engineering*. 2010; 2010. <https://doi.org/10.1155/2010/639801>
46. Podlubny I. *Fractional differential equations*, vol. 198 of *Mathematics in Science and Engineering*; 1999.
47. Figueiredo Camargo R, Chiacchio AO, Capelas de Oliveira E. Differentiation to fractional orders and the fractional telegraph equation. *Journal of Mathematical Physics*. 2008; 49(3):033505. <https://doi.org/10.1063/1.2890375>
48. Waseem W, Sulaiman M, Alhindi A, Alhakamy H. A soft computing approach based on fractional order DPSO algorithm designed to solve the corneal model for eye surgery. *IEEE Access*. 2020;. <https://doi.org/10.1109/ACCESS.2020.2983823>
49. Bukhari AH, Raja MAZ, Sulaiman M, Islam S, Shoaib M, Kumam P. Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting. *IEEE Access*. 2020;. <https://doi.org/10.1109/ACCESS.2020.2985763>
50. Khan A, Sulaiman M, Alhakami H, Alhindi A. Analysis of oscillatory behaviour of heart by using a novel neuroevolutionary approach. *IEEE Access*. 2020; p. 1–1.
51. Bukhari AH, Raja MAZ, Sulaiman M, Islam S, Shoaib M, Kumam P. Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting. *IEEE Access*. 2020; 8:71326–71338. <https://doi.org/10.1109/ACCESS.2020.2985763>
52. Bukhari AH, Sulaiman M, Islam S, Shoaib M, Kumam P, Raja MAZ. Neuro-fuzzy modeling and prediction of summer precipitation with application to different meteorological stations. *Alexandria Engineering Journal*. 2020; 59(1):101–116. <https://doi.org/10.1016/j.aej.2019.12.011>
53. Pires ES, Machado JT, de Moura Oliveira P, Cunha JB, Mendes L. Particle swarm optimization with fractional-order velocity. *Nonlinear Dynamics*. 2010; 61(1-2):295–301. <https://doi.org/10.1007/s11071-009-9649-y>
54. Couceiro M, Ghamisi P. Particle swarm optimization. In: *Fractional Order Darwinian Particle Swarm Optimization*. Springer; 2016. p. 1–10.
55. Waseem W, Sulaiman M, Islam S, Kumam P, Nawaz R, Raja MAZ, et al. A study of changes in temperature profile of porous fin model using cuckoo search algorithm. *Alexandria Engineering Journal*. 2020; 59(1):11–24. <https://doi.org/10.1016/j.aej.2019.12.001>