

Numerical solution for non-linear differential equation of Emden-Fowler by artificial neural network

Nahdh S. M. Al-Saif ^{1*}

¹Department of Mathematics, College of Science, University of Anbar, Iraq

ARTICLE INFO

Received: 18/10/2025
Revised: 12/01/2026
Accepted: 01/02/2026
Available online: 08/08/2026
August Issue
[10.37652/juaps.2026.166401.1789](https://doi.org/10.37652/juaps.2026.166401.1789)

 CITE @ JUAPS

Corresponding author

Nahdh S. M. Al-Saif
sc.dr.nahdhal-ani@uoanbar.edu.iq

ABSTRACT

The Emden - Fowler equation ($E^- F^-$ Eq) is used in math and other sciences like chemistry, astrophysics and physics. It can also be changed to become the Lane- Emden equation ($L^- E^-$ Eq) by using a specific function, and it is applied in various scientific fields along with math. Many researchers examine different ways, both analytical and numerical, to solve these types of equations, whether they are linear or nonlinear. Using artificial intelligence and machine learning, we can now find solution to complicated equations more quickly and accurately than with traditional methods. The goal of this study is solution ($E^- F^-$ Eq) singular second order differential equations via neural network by three main steps. First, a neural network (A-N-N) setup (a type of algorithm) is created to mimic the problem being worked on. Next, this algorithm is used to find a numerical solution for ($E^- F^-$ Eq). Finally, the results are compared with well-known approximate answers from various examples, using tables and figures for specific parameters to show the accuracy and effectiveness of the ANN method.

Keywords: Artificial Neural Networks (A-N-N), Emden-Fowler equation ($E^- F^-$ Eq), LM Training, Tansig (tanh) function

1 INTRODUCTION

Generalized ($E^- F^-$ Eq) appears in many real-life applications in engineering and science. It arises in the study of stellar structure and galaxy clusters. In astrophysics, the ($E^- F^-$ Eq) can be considered a simplified form of Poisson's equation and is used to describe the gravitational behavior of a self-gravitating spherical gas cloud.

The ($L^- E^-$ Eq), or Lane-Emden equation, is named after Robert Emden and Jonathan Homer Lane, who contributed to the study of stellar structure and astrophysical systems. The ($L^- E^-$ Eq) describes the density distribution of a gaseous sphere under specific physical assumptions, such as spherical symmetry and, in some cases, constant temperature.

In general, the ($E^- F^-$ Eq) has the following form

[1, 2]:

$$V'''(x) + \frac{\gamma}{x}V'(x) + \mathcal{F}(x)\mathcal{G}(V) = H(x), \quad x > 0 \quad (1)$$

with

$$V(0) = \eta, \quad V'(0) = m,$$

where η , m , and γ are numbers, and $\mathcal{F}(x)$, $\mathcal{G}(V)$, and $H(x)$ are given functions.

Because the ($E^- F^-$ Eq) is important and has many applications, several analytical, numerical, and semi-analytical methods have been developed to solve it. Different approaches have been proposed over the years to obtain approximate or exact solutions for linear and nonlinear forms of this equation. Power-series methods and the Homotopy Analysis Method (HAM) have been used to obtain step-by-step solutions of the ($E^- F^-$ Eq) [1]. The Homotopy Perturbation Method (HPM) has also been

applied to derive simple and accurate solutions for generalized Emden-Fowler equations, which are second-order ordinary differential equations (ODEs) [2]. In addition, a hybrid method combining the variational iteration method and HPM has been proposed as an effective approach for solving the $(E^- F^- \text{ Eq})$ [3]. Another efficient technique, known as SGIPSM, has been introduced to solve the $(E^- F^- \text{ Eq})$ with mixed Neumann and Robin boundary conditions, providing accurate results within a short computational time [4].

Other studies have used matrix and polynomial-based methods. A first-kind Chebyshev differentiation matrix has been applied to solve linear and nonlinear forms of the $(E^- F^- \text{ Eq})$ [5]. The Taylor series method has also been presented as a simple and accessible technique for solving related problems [6]. In addition, a Morlet wavelet neural network (MWNN), optimized using a genetic algorithm (GA) and an interior-point algorithm (IPA), has been developed to solve second-order $(E^- F^- \text{ Eq})$ problems [7]. The Finite Element Method (FEM), using both (p)- and (h)-approaches, together with the Adomian Decomposition Method (ADM), has also been used to solve the $(E^- F^- \text{ Eq})$ [8]. Furthermore, the advanced Adomian decomposition method has been applied to obtain approximate solutions for special initial-value problems related to the $(E^- F^- \text{ Eq})$ [9]. Fractional forms of the $(E^- F^- \text{ Eq})$ have also been considered, with computational solutions obtained using Boubaker polynomials [10]. In [11], a numerical solution of the generalized $(E^- F^- \text{ Eq})$ was obtained using the Galerkin method. In [12], the Laplace Transform Homotopy Perturbation Method (LT-HPM) and the Elzaki Transform Homotopy Perturbation Method (ET-HPM) were combined to solve the considered $(E^- F^- \text{ Eq})$. Similarly, in [13], a method combining HAM and the variational iteration method was proposed for solving the $(E^- F^- \text{ Eq})$.

Recently, neural networks have become increasingly important in numerical analysis. They have been used to develop computational algorithms for solving mathematical problems involving integration and differentiation. The use of neural networks in calculus provides new possibilities for solving complex mathematical problems and has potential applications in engineering, physics, and applied mathematics. Because neural networks have strong approximation capabilities, they can be useful for solving different boundary-value and initial-value problems. In addition, neural-network-based methods may offer advantages over traditional mathematical methods, as they can approximate unknown functions without

requiring lengthy analytical calculations, and the computational complexity does not necessarily increase sharply when more points are used.

The (ANN) method can be applied to solve both simple and complex ordinary and partial differential equations. For example, feed-forward neural networks have been used to solve partial differential equations [14]. Two simple approaches based on an (ANN) with three layers have also been proposed to solve the $(E^- F^- \text{ Eq})$ [15]. Another study suggested using an (ANN) to solve important classes of $(E^- F^- \text{ Eq})$ problems by constructing Chebyshev and Legendre blocks. This efficient neural-network structure is based on Jacobi polynomials and is designed to approximate solutions of complex differential equations [16]. In [17], the wave-equation problem was transformed into a standard form, and a neural-network procedure based on linear regression was used to obtain an accurate approximate solution. The main objective of this study is to solve singular second-order $(E^- F^- \text{ Eq})$ differential equations using a neural-network approach through three main steps. First, an (A-N-N) structure is constructed to represent the mathematical problem under consideration. Second, this algorithm is used to obtain a numerical solution of the $(E^- F^- \text{ Eq})$. Finally, the obtained results are compared with well-known approximate solutions from selected examples. Tables and figures for specific parameter values are used to demonstrate the accuracy and effectiveness of the (A-N-N) method.

2 ARTIFICIAL NEURAL NETWORK METHOD (ANN)

2.1 Structure of ann

The early efforts to create (ANN) were inspired by the goal of making models similar to how natural brains work. Much later, people found out that (ANN) are a flexible way to estimate probabilities based on a group of samples (the input data). The main part of an (ANN) is the neuron. A neuron is a unit that takes in several inputs and has one output. First, each input x_i is multiplied by a weight ω_i , and then we add everything together to get a total, which equals θ , ($\sum x_i \omega_i = \theta$). Next, an activation function is used on the result θ . The output from the neuron is considered to be $f(\theta)$. The network has many layers of tiny brain cells called neurons. The first layer (called layer 0) sends the input data to the next layer, which is layer 1. Layer 0 doesn't do any processing; you can think of it as a layer that senses things. Each neuron

takes a piece of the input, which is a set of numbers, and sends this piece exactly as it is to all the neurons in the input layer. the last layer is the result. A layer that provides the final answer; each output from the neurons contributes to the overall result, which is the output vector y . The layers that sit between the output and the input called hidden layer [18].

Notes

1. Layer 0 should contain the same number of neurons as the number of components in the input vector (x).
2. The output layer should contain the same number of neurons as the number of required outputs. This number depends on the dimension of the output vector (y).
3. The hidden layers and input layer may contain different numbers of neurons, depending on the structure required to obtain suitable results for a specific problem.

In a feedforward network, each neuron receives input from all neurons in the previous layer, except for the first sensory layer [18].

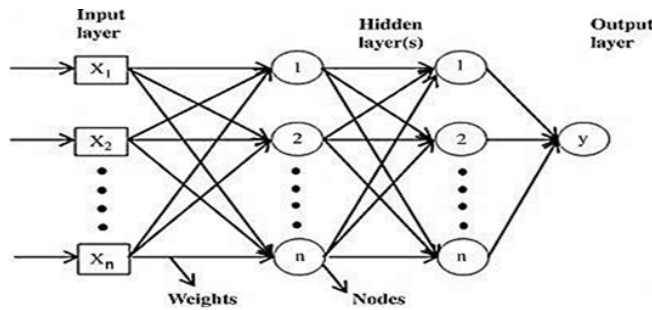


Fig. 1 Structure of ANN

2.2 Hyperbolic-tangent

The activation function commonly used as an alternative to the sigmoid function is the hyperbolic tangent function. This function is mathematically related to the sigmoid function and can be considered a scaled and shifted form of it [19].

$$f(r) = \tanh(r) = \frac{e^r - e^{-r}}{e^r + e^{-r}}$$

2.3 levenberg-marquardt algorithm(lm_algorithm)

This algorithm is made specifically for the sum-of-squares error function. Let ε_l be the error from the L -th training model, and let $\varepsilon^T = (\varepsilon_1 \dots \varepsilon_l)$. The errors has shape

$$E_t = \frac{1}{2} \sum_{l=1}^L (\varepsilon_l)^2 \quad (2)$$

Let's $\forall$ look at the Jacobian matrix. If we think about a small change in the weights from step t to step $t + 1$, we can express the error vector ε using the first part of a Taylor series.

$$\varepsilon_{(t+1)} = \varepsilon_t + \forall (\omega_{t+1} - \omega_t)$$

t calculates error at step $t + 1$ is:

$$E_{(t+1)} = \frac{1}{2} \|\varepsilon_t + \forall (\omega_{t+1} - \omega_t)\|^2 \quad (3)$$

Minimize (3) means:

$$\begin{aligned} \nabla E_t|_{\omega_{t+1}} &= [\varepsilon_t + \forall (\omega_{t+1} - \omega_t)] \forall^T = 0 \\ \rightarrow \varepsilon_t + \forall (\omega_{t+1} - \omega_t) &= 0 \end{aligned}$$

The matrix $\forall$ is not square, so we start by multiplying it on the left by its transpose ($\forall^T$). After that, we multiply by the inverse of the product ($\forall^T \forall$) on the left. This gives us the final result:

$$\omega_{t+1} = \omega_t - \left(\forall^T \forall \right)^{-1} \forall^T \varepsilon_t \quad (4)$$

This shows the main part of the Levenberg-Marquardt weights update formula.

One problem to solve is that the formula (4) can produce big numbers for $\nabla \omega$. Sure, please provide the text you would like me to rewrite in simple words. So big that the (3) estimate no longer works. To prevent this problem, you can use the following updated error function instead [19].

$$E_{t+1} = \frac{1}{2} \|\varepsilon_t + \forall (\omega_{t+1} - \omega_t)\|^2 + \delta \|\omega_{t+1} - \omega_t\|^2$$

where δ is a value that controls the size of the $\nabla \omega$. In the same way as for (4) the new the formula is updated to:

$$\omega_{t+1} = \omega_t - \left(\forall^T \forall + \delta I \right)^{-1} \forall^T \varepsilon_t \quad (5)$$

2.4 Validation of models

To check how well the machine learning models work in this study, we used a method called mean square error (MSE). The MSE is used to check how much the real values differ from the predicted values. In simple terms, lower MSE values mean the models are more accurate and perform better. The value of MSE is calculated using this formula: [20]

$$MSE = \frac{1}{N} \left(\sum_{k=1}^k (d_k - O_k)^2 \right)$$

where d_k is the actual values and O_k are the predicted values of pattern.

2.5 Methodology

In this study, the proposed method includes the following steps:

- a Data collection: This is the first step and involves collecting data from previous studies, research papers, and related articles. The collected data are divided into two parts: a training dataset and a testing dataset. The training dataset represents

70% of the total data, whereas the testing dataset represents 30%.

- b Model building: In this step, the training dataset is used to train the model using suitable training algorithms, such as the Levenberg-Marquardt algorithm.
- c Model validation: In the final step, the testing dataset is used to evaluate the performance and accuracy of the trained model. Statistical measures, such as the mean square error (MSE), are used to assess the model performance.

3 RESULTS AND DISCUSSION

In this study, MATLAB tools for (ANN), including the feed-forward neural network (FFNN) module, were used to construct the FFNN model. The selection of the number of layers and neurons, together with the training method, such as TANSIG or L-M, was made to achieve a suitable balance between accuracy and model simplicity. The performance of the neural network was evaluated by comparing the predicted results with the expected results in order to assess its accuracy and effectiveness, as shown in Figure 2.

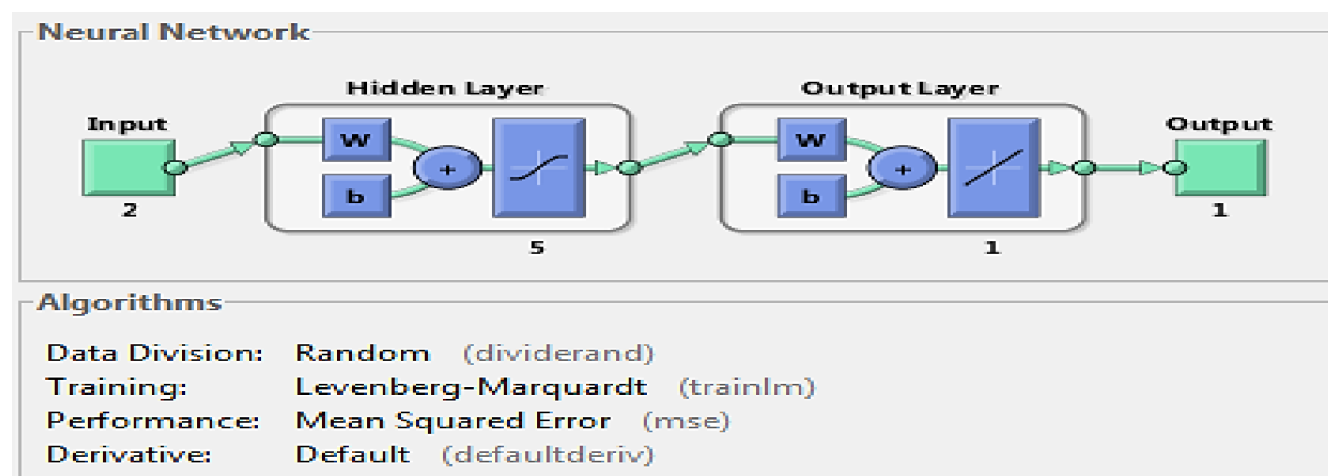


Fig. 2 Neural network and algorithm

The approximate function of the structure has the form:

$$\text{Result} = \text{output} = (\omega_2 * \tanh((\omega_1 * x] + b_1) + b_2)$$

Where ω_1, ω_2 are wights, b_1, b_2 are bais and tanh

tanh is activation function . Now applying the (FFNN) structuration to some problems to calculates the accurse and efficient.

Example 1 let $(E^- F^- E q)$ has shape

$$V''(\kappa) + \frac{2}{x}V'(\kappa) + e^V = 0 \quad \kappa > 0 \quad (6)$$

with

$$V(0) = 0, V'(0) = 0$$

and actual solution

$$V(x) = -\frac{1}{6}x^2 + \frac{1}{120}x^4 + \frac{1}{1890}x^6$$

We applied the A_N_N method on equation (6) and compared result with the sources [9, 12].

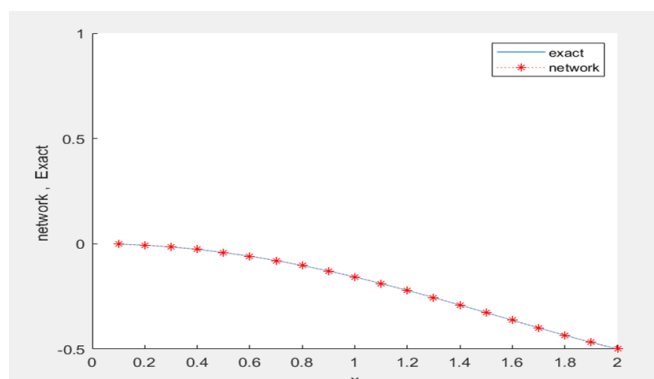


Fig. 3 Comparison of $V(x)$ values of (E^-F^-Eq) with exact solution Example 1

Example 2 Let (E^-F^-Eq) has shape

$$V''(x) + \frac{2}{x}V'(x) + 1 = 0 \quad x > 0 \quad (7)$$

with

$$V(0) = 1, V'(0) = 0$$

and actual solution

$$V(x) = 1 - \frac{x^2}{6}$$

We applied the A-N-N method to solve equation (7) and compared result with the source [16].

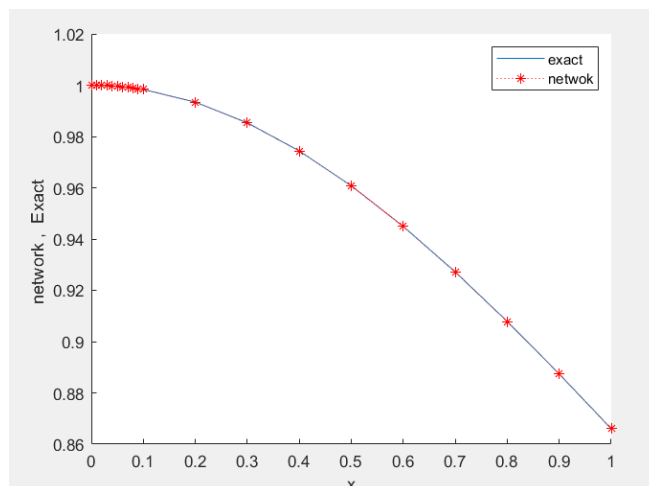


Fig. 4 Comparison of $V(x)$ values of (E^-F^-Eq) with exact solution example 2.

example 3 let (E^-F^-Eq) has shape

$$V''(\kappa) + \frac{2}{x}V'(\kappa) + V(\kappa) = 0 \quad \kappa > 0 \quad (8)$$

with

$$V(0) = 1, V'(0) = 0$$

and actual solution

$$V(x) = \frac{\sin x}{x}$$

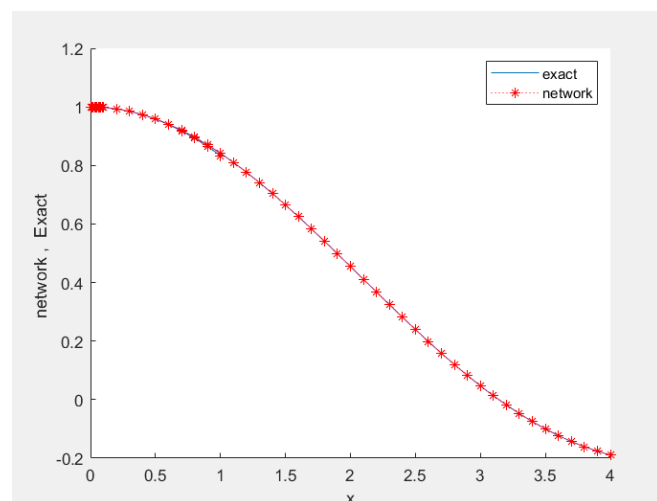
We applied the A-N-N method to solve equation (8) and compared result with the source [16].

Table 1 Comparing the $V(x)$ values for (E'F. Eq) in example 1.

$X \downarrow$	Actual $V(x)$	Error [12] $v4(t)$	Error [9]	$V(x)$ ANN	Error ANN
0.1	-1.665832804e-03	3.07854e-12	2.22006e-06	-1.665829261e-03	3.54225885e-09
0.2	-6.653299470e-03	7.85679e-10	3.54178e-05	-6.653667106e-03	3.67635396e-07
0.3	-1.493211428e-02	2.00348e-08	1.78437e-04	-1.493217841e-02	6.41304800e-08
0.4	-2.645116613e-02	1.98721e-07	5.60158e-04	-2.645108133e-02	8.48037939e-08
0.5	-4.113756613e-02	1.17389e-06	1.35583e-03	-4.113753895e-02	2.71823918e-08
0.6	-5.889531428e-02	4.99298e-06	2.78219e-03	-5.889537556e-02	6.12788116e-08
0.7	-7.960358518e-02	1.69206e-05	5.09154e-03	-7.960364421e-02	5.90260221e-08
0.8	-1.031146328e-01	4.85356e-05	8.56513e-03	-1.031146083e-01	2.44613243e-08
0.9	-1.292513142e-01	1.22531e-04	1.35060e-02	-1.292512166e-01	9.76185565e-08
1.00	-1.578042328e-01	2.79618e-04	2.02319e-02	-1.578041524e-01	8.03964700e-08
MSR		1.70e-14		1.70e-14	

Table 2 Comparing the $V(x)$ values for (E⁻F⁻Eq) in example 2.

$x \downarrow$	Exact	Legendre and Chebyshev blocks Method [16]		Present method	
		$V(x)$	Error	V-ANN	Error-ANN
0.00	1.000000000e+00	0.999988760931	1.12e-05	1.0000000544620e+00	5.4461990917076e-08
0.01	9.999833333e-01	0.999973834547	9.50e-06	9.9998333267860e-01	6.5473682031580e-10
0.02	9.999333333e-01	0.999924811203	8.52e-06	9.9993330146728e-01	3.1866051575236e-08
0.05	9.995833333e-01	0.999574749438	8.58e-06	9.9958329875224e-01	3.4581096985953e-08
0.10	9.983333333e-01	0.998321196625	1.21e-05	9.9833339047025e-01	5.7136914977107e-08
0.50	9.583333333e-01	0.958322307305	1.10e-05	9.583340701296e-01	7.3679625844925e-08
1.00	8.333333333e-01	0.833325953763	7.38e-06	9.583340701296e-01	4.9575599181395e-09
MSR		1.04e-10		2.09e-15	

**Fig. 5** Comparison of $V(x)$ values of (E⁻F⁻Eq) with exact solution example 3

example 4 let (E⁻F⁻Eq) has shape

$$V''(x) + \frac{2}{x}V'(x) + V^5(x) = 0 \quad x > 0 \quad (9)$$

with

$$V(0) = 1, V'(0) = 0$$

and exact solution

$$V(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}$$

We applied the A-N-N method to solve equation (9) and compared result with the source [16].

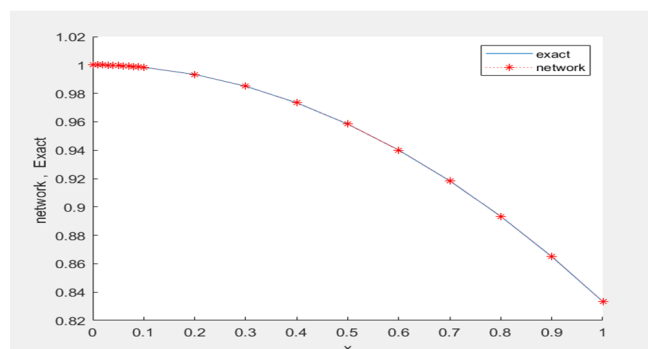
**Fig. 6** Comparing of $V(x)$ values of (E⁻F⁻Eq) with exact solution example 4.

Table 3 Comparing the $V(x)$ values for (E^-F^-Eq) in example 3.

$X \downarrow$	Exact	Legendre and Chebyshev blocks Method [16]		Present method	
		$V(x)$	Error	V-ANN	Error-ANN
2.00	0.454648713412	0.454685572466	3.690e-05	4.5464871769954e-01	4.2866978300360e-09
2.20	0.367498365372	0.367519111023	2.07e-05	3.6749775238091e-01	6.1299163112150e-07
2.40	0.281442991896	0.281443512178	5.20e-07	2.8144227866209e-01	7.1323422456970e-07
2.60	0.198269758392	0.198267411193	2.35e-06	1.9826953606348e-01	2.2232938950983e-07
2.80	0.119638625055	0.119649029987	1.04e-05	1.1963901218913e-01	3.8713344580055e-07
3.20	-0.01824191982	-0.018230879653	1.10e-05	-1.8242120059702e-02	2.0023858324572e-07
3.40	-0.07515914765	-0.075165760742	6.61e-06	-7.5160091605188e-02	9.4395023793326e-07
3.60	-0.12292234535	-0.122941560015	1.92e-05	-1.2292296867213e-01	6.2331245358205e-07
3.80	-0.16101523445	-0.161035028141	1.98e-05	-1.6101439967361e-01	8.3478500137457e-07
4.00	-0.18920062382	-0.189214194644	1.36e-05	-1.8920089115644e-01	2.6732945518360e-07
MSE		3.07e-10		3.23e-13	

Table 4 Comparing the $V(\kappa)$ values for (E^-F^-Eq) in example 4.

$X \downarrow$	Exact	Legendre and Chebyshev blocks Method [16]		Present method	
		V- [16]	Error- [16]	V-ANN	Error-ANN
0.00	1.000000000000	0.99999126178210	8.740e-06	1.0000001150927e+00	1.1509265651632e-07
0.01	0.999983333749	0.99997609571095	7.24e-06	9.9998332851412e-01	5.2358660740381e-09
0.02	0.999933339999	0.99992723329405	6.11e-06	9.9993326792296e-01	7.2076300927648e-08
0.05	0.999583593569	0.99957909452205	4.50e-06	9.9958352085739e-01	7.2711901721867e-08
0.10	0.998337488459	0.99833189363358	5.59e-06	9.9833760484564e-01	1.1638605523778e-07
0.50	0.960768922830	0.96076180195597	7.12e-06	9.6076909013802e-01	1.6730749785587e-07
1.00	0.866025403784	0.86602176633179	3.64e-06	8.6602541237449e-01	8.5900523361104e-09
MSR		5.87e-11		1.42e-14	

4 CONCLUSION

The main aim of this paper is to present an effective method based on Artificial Neural Networks (ANN) for obtaining approximate solutions to important classes of Lane-Emden equations, which represent a type of singular second-order differential equation. The hidden layers of the proposed network contain blocks that use the tansig, or hyperbolic tangent, activation function. The final block adjusts the network outputs to produce the predicted values.

The results presented in the tables and figures show the performance of the proposed neural-network model by comparing its solutions with previously reported solutions. Compared with other approximate numerical methods, the proposed method based on the tanh activation function provides more accurate results in the considered examples. Therefore, the use of these blocks in different layers of other neural-network structures may be considered in future studies to obtain improved approximate

solutions.

Funding source

No funds received.

Data availability

N/A

DECLARATIONS

Conflict of interest

The authors declare no competing interests

Consent to publish

N/A

Ethical approval

N/A

REFERENCES

- [1] Van Gorder RA, Vajravelu K. Analytic and numerical solutions to the Lane–Emden equation. *Physics Letters A*. 2008;372(39):6060–6065. [10.1016/j.physleta.2008.08.002](https://doi.org/10.1016/j.physleta.2008.08.002)
- [2] Chowdhury MSH, Hashim I. Solutions of Emden–Fowler equations by homotopy-perturbation method. *Nonlinear Analysis: Real World Applications*. 2009;10(1):104–115. [10.1016/j.nonrwa.2007.08.017](https://doi.org/10.1016/j.nonrwa.2007.08.017)
- [3] Singh M, Verma AK. An effective computational technique for a class of Lane–Emden equations. *Journal of Mathematical Chemistry*. 2016;54(1):231–251. [10.1007/s10910-015-0557-8](https://doi.org/10.1007/s10910-015-0557-8)
- [4] Elgindy KT, Refat HM. High-order shifted Gegenbauer integral pseudo-spectral method for solving differential equations of Lane–Emden type. *Applied Numerical Mathematics*. 2018;128:98–124. [10.1016/j.apnum.2018.01.018](https://doi.org/10.1016/j.apnum.2018.01.018)
- [5] Sharma B, Kumar S, Paswan MK, Mahato D. Chebyshev Operational Matrix Method for Lane–Emden Problem. *Nonlinear Engineering*. 2018;8(1):1–9. [10.1515/nleng-2017-0157](https://doi.org/10.1515/nleng-2017-0157)
- [6] He JH, Ji FY. Taylor series solution for Lane–Emden equation. *Journal of Mathematical Chemistry*. 2019;57(8):1932–1934. [10.1007/s10910-019-01048-7](https://doi.org/10.1007/s10910-019-01048-7)
- [7] Sabir Z, Wahab HA, Umar M, Sakar MG, Raja MAZ. Novel design of Morlet wavelet neural network for solving second order Lane–Emden equation. *Mathematics and Computers in Simulation*. 2020;172:1–14. [10.1016/j.matcom.2020.01.005](https://doi.org/10.1016/j.matcom.2020.01.005)
- [8] Hossen KAA. On Solutions of Emden–Fowler Equation. *American Journal of Computational Mathematics*. 2020;10(01):90–99. [10.4236/ajcm.2020.101006](https://doi.org/10.4236/ajcm.2020.101006)
- [9] Umesh, Kumar M. Approximate solution of singular IVPs of Lane–Emden type and error estimation via advanced Adomian decomposition method. *Journal of Applied Mathematics and Computing*. 2020;66(1-2):527–542. [10.1007/s12190-020-01444-2](https://doi.org/10.1007/s12190-020-01444-2)
- [10] Bencheikh A, Chiter L. Boubaker operational matrix method for fractional Emden–Fowler problem. *Journal of Siberian Federal University Mathematics & Physics*. 2023;16(3):357–69
- [11] Aswhad A. Approximate Solution of Emden–Fowler Equation Using the Galerkin Method. *Journal of University of Anbar for Pure Science*. 2023;17(1):200–203. [10.37652/juaps.2023.178798](https://doi.org/10.37652/juaps.2023.178798)
- [12] Luo X. Analytical solution of time-fractional Emden–Fowler equations arising in astronomy and mathematical physics. *Scientific Reports*. 2024;14(1). [10.1038/s41598-024-79163-y](https://doi.org/10.1038/s41598-024-79163-y)
- [13] Sinha VK, Maraju P. An Effective Numerical Algorithm for Solving the Lane–Emden Type Equation Based on the Variational Iteration Method Coupled with the Homotopy Analysis Method. *Journal of Applied Analysis & Computation*. 2025;15(2):1091–106. [10.11948/20240255](https://doi.org/10.11948/20240255)
- [14] Al-Saif NSM. Neural Network for Solving Integro Partial Differential Equation. In: 2022 2nd International Conference on Computing and Machine Intelligence (ICMI). IEEE; 2022. p. 1–4. [10.1109/icmi55296.2022.9873771](https://doi.org/10.1109/icmi55296.2022.9873771)
- [15] Pourbabaee M. The Neural Network Method for Solving the Lane–Emden Equation with Chebyshev Polynomials of the Second Kind. In: Rahmati Z, editor. *Proceedings of the First International Conference on Machine Learning and Knowledge Discovery (MLKD 2024)*; 2024. p. 439–46. Available from: <https://mlkd.aut.ac.ir/proceedings/2024/paper/4B.5.pdf>
- [16] Parand K, Aghaei AA, Kiani S, Zadeh TI, Khosravi Z. A neural network approach for solving nonlinear differential equations of Lane–Emden type. *Engineering with Computers*. 2023;40(2):953–969. [10.1007/s00366-023-01836-5](https://doi.org/10.1007/s00366-023-01836-5)

- [17] Al-Saif NS, Shamran MA, Ali SMA, Al-Azzawi SN. Solution of Wave Equation by Linear Regression Artificial Neural Network. Baghdad Science Journal. 2024. [10.21123/bsj.2024.11520](https://doi.org/10.21123/bsj.2024.11520)
- [18] Hagan MT, Demuth HB, Beale MH, De Jesús O. Neural Network Design. 2nd ed. Martin Hagan; 2014
- [19] Hristev RM. The ANN Book. 1st ed. Self-published; 1998. Electronic book released under the GNU General Public License, version 2
- [20] Nguyen TA, Ly HB, Mai HVT, Tran VQ. On the Training Algorithms for Artificial Neural Network in Predicting the Shear Strength of Deep Beams. Complexity. 2021;2021(1). [10.1155/2021/5548988](https://doi.org/10.1155/2021/5548988)

How to cite this article

Al-Saif NS. Numerical Solution for Non-Linear Differential Equation of Emden-Fowler by Artificial Neural Network.. Journal of University of Anbar for Pure Science. 2026; 20(2):404-412. doi:[10.37652/juaps.2026.166401.1789](https://doi.org/10.37652/juaps.2026.166401.1789)