

Deep neural network solutions to Newell-Whitehead-Segel equations

Soumaya Nouna¹, Ilyas Tammouch², Assia Nouna¹, Mohamed Mansouri¹

¹Laboratory LAMSAD, Department of Mathematics and Informatics, Hassan First University of Settat, Berrechid, Morocco

²Laboratory of Telecommunications Systems and Decision Engineering, Faculty of Science, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Dec 3, 2024

Revised Oct 22, 2025

Accepted Nov 8, 2025

Keywords:

Artificial neural networks

Deep learning

Deep neural network

NeuroDiffEq

Newell-Whitehead-Segel equations

Abstract

In this work, we use the deep neural network (DNN) approach called NeuroDiffEq, and the unified finite difference exponential approach for obtaining the approximated and exact solutions of Newell-Whitehead-Segel systems that are essential for the biology of mathematics. A unified approach was used to generate several solutions for solitary waves of those systems. The approximated solutions for selected studies are explored using the NeuroDiffEq approach, which is the artificial neural networks (ANN) approach and is based upon trial approximate solution (TAS). The comparison between the obtained approximated solutions and the analytical solutions indicates that the applied method has proved an efficient as well as a highly successful approach to solving various types of the Newell-Whitehead-Segel equations.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Soumaya Nouna

Laboratory LAMSAD, Department of Mathematics and Informatics, Hassan First University of Settat
Berrechid, Morocco

Email: s.nouna@uhp.ac.ma

1. INTRODUCTION

A more general form of the Newell-Whitehead-Segel system is as (1):

$$\frac{\partial v}{\partial t} = \beta \frac{\partial^2 v}{\partial x^2} + cv + kv^p, \quad x \in [a, b], \quad t \in [0, T] \quad (1)$$

With an initial condition as (2):

$$v(x, 0) = f(x) \quad (2)$$

And the boundary conditions as (3):

$$v(a, t) = g_1(t), \quad v(b, t) = g_2(t), \quad t \geq 0. \quad (3)$$

Where c, k , and β represent the real numbers, and p represents the integers of positive numbers. Newell-Whitehead-Segel equations [1], [2] described dynamic properties around the point of bifurcation of the Rayleigh-Benard convective flow for the binary fluid mixtures. These equations are found throughout various applications in science, including mathematical biological sciences, quantum mechanical sciences, and the physics of plasma. Different approaches are being proposed for solving Newell Whitehead Segel equations: laplace adomian decomposition approach, cubic B-spline collocation approach, finite difference scheme, He's variational iteration approach, and homotopy perturbation approach [3]-[6]. The problem with these traditional

methods for solving equations is they can fail before achieving the best-approximated solutions, and they are not promising in terms of accuracy. In the absence of discretization processes, artificial neural networks (ANNs) are alternatives.

ANN is extensively implemented for solving several problems in scientific computing [7], [8]. ANN-based techniques have also been developed to address partial differential equations (PDEs). For instance, Malek and Beidokhti [9] combined artificial neural networks with the Nelder–Mead simplex method to obtain accurate numerical approximations of nonlinear PDEs. Furthermore, Sirignano and Spiliopoulos [10] introduced the deep Galerkin method (DGM), a fully mesh-free neural network framework capable of solving high-dimensional PDEs through stochastic training. Building on these advances, Nouna *et al.* [11] demonstrated the effectiveness of deep neural networks in approximating nonlinear equations such as the Klein–Gordon and Sine–Gordon models, while Nouna *et al.* [12] extended these concepts to a broader class of PDEs, confirming the robustness and versatility of ANN-based solvers for scientific computing. In parallel, another neural-network-based strategy, the physics-informed neural network (PINN), was introduced by Raissi *et al.* [13], and further enhanced by Guo *et al.* [14] through residual-based adaptive refinement mechanisms.

ANNs' ability to solve problems in PDEs has certain benefits, such as differentiable and continuous approximated solutions, strong interpolative features, and reduced memories. Further benefits of the ANN include the ability to use automatically differentiating utilities, for example, PyTorch [15] and TensorFlow [16], which allows the researchers to develop more straightforward approaches for solving the problems of PDE [17]. In this research, we present a methodology to resolve the Newell–Whitehead–Segel models using a model of ANN known as NeuroDiffEq, as the ANN approximator having trial approximate solution (TAS) applied [18].

The structure of this paper is as follows: in section 2, we provide the exact solutions to the Newell–Whitehead–Segel models. Section 3 presents the NeuroDiffEq approach for solving these systems. Numerical solutions for various Newell–Whitehead–Segel types using the NeuroDiffEq approach are introduced and compared with their exact solutions in section 4. Finally, in section 5, we conclude our work.

2. EXACT SOLUTIONS

In this section, we describe some concepts in a unified approach [19]–[21] to finding some analytical solutions to by (1). To find the wave solutions, the transformation as in (4) is used:

$$v(x, t) = v(\kappa), \quad \kappa = x - \varepsilon t \quad (4)$$

In (1), the result in (5) is obtained:

$$cv + kv^p + \varepsilon v' + \beta v'' = 0 \quad (5)$$

with $v' = \frac{dv}{d\kappa}$. In (5) can be integrated if $p = 2$ or $p = 3$. Therefore, we restrict the search to the solutions of (1) in both cases.

2.1. First case

In the case of $p = 2$, the (5) would have a form as (6):

$$cv + kv^2 + \varepsilon v' + \beta v'' = 0 \quad (6)$$

A unified approach states that the requisite solutions are expressed in the following terms as in (7):

$$v(\kappa) = \sum_{i=0}^m q_i \Phi^i(\kappa) \quad (7)$$

Auxiliary feature $\Phi(\kappa)$ fulfills the following auxiliary equation as in (8):

$$\Phi'(\kappa) = \sum_{i=0}^n k_i \Phi^i(\kappa) \quad (8)$$

with q_i and k_i being the real constants determined at some later. The homogeneous equilibrium between v'' and v^2 in (6) gives $m = 2(n - 1)$, $n \geq 1$. The exact solutions of the waves for (1) are found here when $n = 2$. According to (7) and (8), we get (9) and (10):

$$v(\kappa) = q_0 + q_1 \Phi(\kappa) + q_2 \Phi^2(\kappa) \quad (9)$$

$$\Phi'(\kappa) = k_0 + k_1\Phi(\kappa) + k_2\Phi^2(\kappa). \quad (10)$$

By substituting (9) with (10) in (6) and setting $\Phi(\kappa)$'s coefficients equal to null, the solutions of the parameters q_i and k_i , $i = 0, 1, 2$ have the following form as in (11):

$$q_0 = -\frac{3(k_1 + \lambda)^2\beta}{2k}, q_1 = -\frac{6k_2(k_1 + \lambda)\beta}{k}, q_2 = -\frac{6k_2^2\beta}{k}, \quad (11)$$

$$\varepsilon = 5\lambda\beta, \quad c = 6\lambda^2\beta, \quad \lambda = \sqrt{k_1^2 - 4k_0k_2}.$$

By using (11) in (9) and solving auxiliary equation provided from (10), the solution to (1) is obtained as (12):

$$v_1(x, t) = -\frac{3\lambda^2\beta}{2k} \left(1 - 2 \tanh\left(\frac{\lambda}{2}\kappa\right) + \tanh^2\left(\frac{\lambda}{2}\kappa\right) \right) \quad (12)$$

with $\kappa = x - \varepsilon t$ and $v_1(x, t)$ is the exact solution.

2.2. Second case

In the case of $p = 3$, the (5) would have a form as (13):

$$cv + kv^3 + \varepsilon v' + \beta v'' = 0 \quad (13)$$

Considering a homogeneous equilibrium between v'' and v^3 in (13), we obtain $m = n - 1$, $n \geq 1$. Likewise, the exact solutions of the waves to (1) are found when $n = 2$. According to (7) and (8), we get (14) and (15):

$$v(\kappa) = q_0 + q_1\Phi(\kappa) \quad (14)$$

$$\Phi'(\kappa) = k_0 + k_1\Phi(\kappa) + k_2\Phi^2(\kappa). \quad (15)$$

By substituting (14) with (15) in (13) and setting $\Phi(\kappa)$'s coefficients equal to null, the solutions of the parameters q_i and k_i have the (16) form:

$$q_0 = \frac{(k_1 + \lambda)\sqrt{-\beta}}{\sqrt{2k}}, q_1 = \frac{k_2\sqrt{-\beta}}{\sqrt{k}}, \quad \varepsilon = 2\lambda\beta, \quad c = 2\lambda^2\beta, \quad \lambda = \sqrt{k_1^2 - 4k_0k_2}, \quad \beta > 0. \quad (16)$$

By using (16) in (14) and solving auxiliary equation provided from (15), the solution to (1) is obtained as (17):

$$v_2(x, t) = -\frac{\lambda\sqrt{-\beta}}{\sqrt{2k}} \left(1 + \tanh\left(\frac{\lambda}{2}\kappa\right) \right) \quad (17)$$

with $\kappa = x - \varepsilon t$ and $v_2(x, t)$ is the exact solution.

3. ARTIFICIAL NEURAL NETWORK SOLUTIONS

This section introduces the ANN architecture. Followed by the NeuroDiffEq method which uses the ANN architecture. To approximate solutions for Newell Whitehead Segel equations.

3.1. The artificial neural network

ANN comprises $R + 1$ layers, where 0 layer represents an input layer while R layer represents an output layer. $0 < r < R$ layers represent hidden layers. Activation functions inside hidden layers are any activation function, like rectified linear units, hyperbolic tangents, and sigmoids. By default, we will use sigmoid functions (see definition 3.1.) for hidden layers, which possess universal approximation capability (see theorem 3.1.) [22]. Every node of an ANN is biased, which includes output and not input nodes, while connections between nodes in the following layers are presented as matrices of weights (see Figure 1).

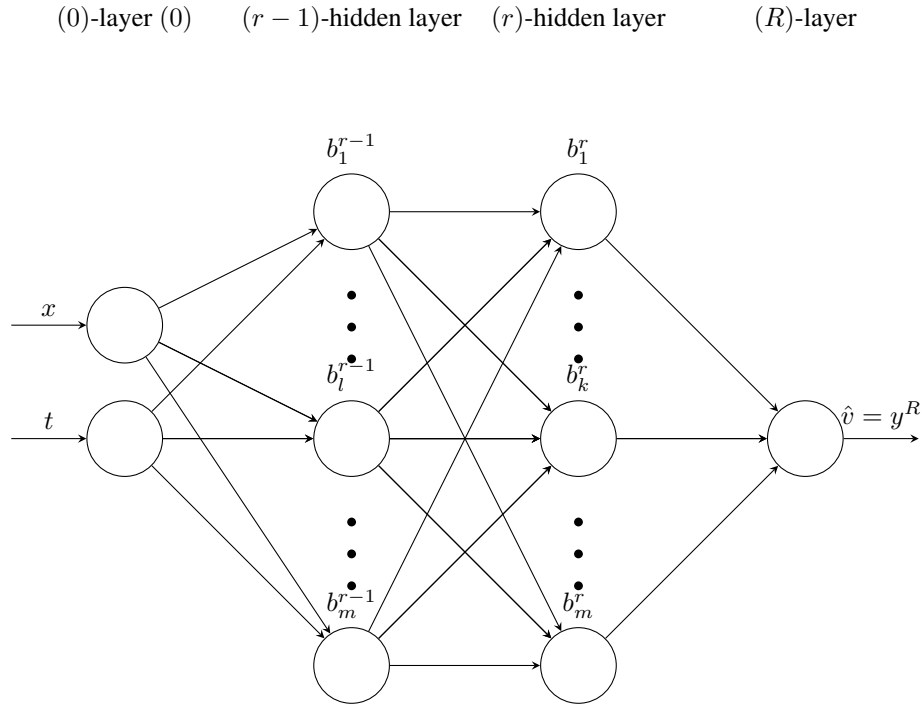


Figure 1. ANN architecture

Let b_k^r represent a bias for neuron k at layer r . w_{kl}^r represent the weights from neuron l at layer $r - 1$ to neuron k at layer r . The r -layer activation function is denoted as φ_r . For the output of neuron k at layer r , we denote it as y_k^r . The important quantity that is widely utilized is known as the weighted input, defined as (18):

$$h_k^r = \sum_l w_{kl}^r \varphi_{r-1}(h_l^{r-1}) + b_k^r \quad (18)$$

Where the summation is computed on every input of the neuron k inside the r -layer. Namely number of neurons in $(r - 1)$ -layer. Weighted entry (18) can obviously be described as an output of the preceding layer in the (19) way:

$$h_k^r = \sum_l w_{kl}^r y_l^{r-1} + b_k^r \quad (19)$$

Where output $y_l^{r-1} = \varphi_{r-1}(h_l^{r-1})$ represents the weighted input activation. Since we are working on ANNs, we prefer the (18) since this naturally describes the recursion of previous weighted inputs into the ANN. As a definition, we have the (20):

$$\varphi_0(h_k^0) = y_k^0 = x_k \quad (20)$$

End all recursion. If we remove subscripts, it is possible to write (18) as a convenient vector expression as (21):

$$h^r = W^r \varphi_{r-1}(h^{r-1}) + b^r = W^r y^{r-1} + b^r \quad (21)$$

in which every component of the vectors h^r and y^r is given respectively by h_k^r and y_k^r , with an activation function applied per component. The matrix elements W^r can be given using $W_{kl}^r = w_{kl}^r$. Using any of the

above definitions, a feed-forward algorithm to calculate output y^R , based on input x , can be given as (22):

$$\left\{ \begin{array}{l} h^1 = W^1 x + b^1 \\ h^2 = W^2 \varphi_1(h^1) + b^2 \\ h^3 = W^3 \varphi_2(h^2) + b^3 \\ \vdots \\ h^{R-1} = W^{R-1} \varphi_{R-2}(h^{R-2}) + b^{R-1} \\ h^R = W^R \varphi_{R-1}(h^{R-1}) + b^R \\ \hat{v} = y^R = \varphi_R(h^R). \end{array} \right. \quad (22)$$

The efficiency of output \hat{v} can be determined by the calculation of the loss function. Consequently, the last step is to minimize the loss function with the stochastic gradient descent (SGD) optimizer. In addition, the output $\hat{v}(x, t, \theta)$ of the ANN is the approximate of $v(x, t)$ in the (1)-(3) where x and t represent inputs, and θ represents the weights and biases of the adjustable parameters. Then we can easily express n th derivatives for $\hat{v}(x, t, \theta)$ via automatic differentiation (AD) (1) as (23) and (24):

$$\frac{\partial^n \hat{v}}{\partial t^n}(x, t, \theta) = \frac{\partial^n \varphi_R(h^R)}{\partial t^n}, \quad n = 1, \quad (23)$$

$$\frac{\partial^n \hat{v}}{\partial x^n}(x, t, \theta) = \frac{\partial^n \varphi_R(h^R)}{\partial x^n}, \quad n = 1, 2. \quad (24)$$

Definition 3.1.: a Sigmoid function [23] can be defined as a function of mathematics that associates its input with a value in the range of 0 to 1, which produces a curve in the shape of an S. However, a sigmoid function is defined mathematically as (25):

$$\varphi(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \quad (25)$$

Where $\varphi(x)$ represents a sigmoid function on input x .

Theorem 3.1.: assume that φ is bounded, strictly monotonic, and increasing. Consider Ω as an m -dimensional compact sub-set by \mathbb{R}^M . The space of continuous real functions which are defined in Ω is denoted as $\mathbf{C}(\Omega)$. So, for any given function h , $h : \Omega \rightarrow \mathbb{R}$, some real $\varepsilon > 0$, and $x \in \Omega$, then there exists a set of real values w_k , b_k , b_l , and w_{kl} , so that we can define (26):

$$H(x) = \sum_k w_k \varphi(\sum_l w_{kl} x_l + b_l) + b_k \quad (26)$$

As the approximate implementation for a function h , as depicted in (27):

$$|H(x) - h(x)| \leq \varepsilon. \quad (27)$$

3.2. The NeuroDiffEq approach

The NeuroDiffEq approach uses ANN to solve the Newell-Whitehead-Segel equations (1)-(3). This approach differs in several aspects, including the generation of data points, the definition of boundary conditions, and the choice of loss functions. Furthermore, to solve these equations with NeuroDiffEq, the primary concept involves constructing the TAS, denoted as $\tilde{v}(x, t; \theta)$ [18]. The NeuroDiffEq algorithm defines in the following ways:

- i) An $(M \times N)$ -grid point set $E = (x, t)$, with $x = (x_1, \dots, x_M)$, $t = (t_1, \dots, t_N)$ within the $[a, b] \times [0, T]$ domain, supplied by the ANNs. Any distribution you choose, such as uniform distribution and equal spacing, can be used to generate points.
- ii) Development of the TAS, a well-known function, is composed of input E and output $\hat{v}(x, t; \theta)$. Moreover, TAS is defined in the (28) form [24]:

$$\tilde{v}(x, t; \theta) = B(x) + G(x, t; \hat{v}) \quad (28)$$

$A(x)$ has been chosen to respect boundary conditions, and $G(x, t; \hat{v})$ has been selected as zero for all (x, t) at the boundary. This gives the TAS automatically satisfying boundary conditions, whatever the output of the ANN. However, this method is also similar in concept to the trial function [25], although the trial function takes a different form. It is also possible to modify the TAS for the initial conditions (2). Generally, solutions that are transformed have the (29) form [18]:

$$\tilde{v}(x, 0; \theta) = v(x, 0) + xt(1-x)(1-t)\hat{v}(x, t; \theta). \quad (29)$$

iii) The rial approximate solution has been implemented to minimize the (30) loss function:

$$\mathcal{L}(\theta) = \mathcal{L}_1(\theta) + \mu\mathcal{L}_2(\theta). \quad (30)$$

The first term of \mathcal{L} can be represented as (31):

$$\mathcal{L}_1(\theta) = \left[\frac{\partial \tilde{v}}{\partial t}(x, t; \theta) - \beta \frac{\partial^2 \tilde{v}}{\partial x^2}(x, t; \theta) - c\tilde{v}(x, t; \theta) - k\tilde{v}^p(x, t; \theta) \right]_{(x,t) \in E}^2 \quad (31)$$

Where (31) represents an approximate solution to the equation itself, with E consisting of the finite points in the equation's domain, then the second term is defined as (32):

$$\mathcal{L}_2(\theta) = [\tilde{v}(x, 0; \theta) - f(x)]_{x \in [a,b]}^2 + [\tilde{v}(a, t; \theta) - g_1(t)]_{t \in [0,T]}^2 + [\tilde{v}(b, t; \theta) - g_2(t)]_{t \in [0,T]}^2 \quad (32)$$

\mathcal{L}_2 satisfies both the initial and boundary conditions. A weighting coefficient called μ indicates the importance of both error components. This factor is arbitrarily determined in practice. A description of the algorithm for this method is given in Algorithm 1.

Algorithm 1 NeuroDiffEq-Algorithm to solve the Newell-Whitehead-Segel equation

1 : Consistently generate 150×150 points $E = \{x_n, t_n\}_{n=1}^{150}$ within the $[0, 1]$ domain.

2 : While $iter < 0$ iteration Do

3 : For every $(x, t) \in E$ Do

4 : To Calculate

$$\hat{v} = (x, t, \theta) = \varphi_R(h^R), \quad \frac{\partial \hat{v}}{\partial t}(x, t, \theta) = \frac{\partial \varphi_R(h^R)}{\partial t}, \quad \frac{\partial^2 \hat{v}}{\partial x^2}(x, t, \theta) = \frac{\partial^2 \varphi_R(h^R)}{\partial x^2}$$

5 : Construct the TAS that is satisfied by initial condition of 2 in the following way

$$\tilde{v}(x, 0; \theta) = v(x, 0) + xt(1-x)(1-t)\hat{v}(x, t; \theta)$$

6 : Calculate a loss function as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_1(\theta) + \mu\mathcal{L}_2(\theta),$$

While

$$\mathcal{L}_1(\theta) = \left[\frac{\partial \tilde{v}}{\partial t}(x_n, t_n; \theta) - \beta \frac{\partial^2 \tilde{v}}{\partial x^2}(x_n, t_n; \theta) - c\tilde{v}(x_n, t_n; \theta) - k\tilde{v}^p(x_n, t_n; \theta) \right]^2,$$

$$\mathcal{L}_2(\theta) = [\tilde{v}(x_n, 0; \theta) - f(x_n)]^2 + [\tilde{v}(a, t_n; \theta) - g_1(t_n)]^2 + [\tilde{v}(b, t_n; \theta) - g_2(t_n)]^2.$$

7 : End For

8 : Minimizing the loss, update weights and biases using the SGD-optimizer.

9 : End While

4. THE NUMERICAL RESULTS

Several types of Newell-Whitehead-Segel equations are used to demonstrate the performance of the NeuroDiffEq approach. For all the numerical examples below, we implemented an ANN with five hidden

layers, where the first layer has 20 neurons, the second has 30 neurons, the third has 40 neurons, the fourth has 50 neurons, and the last has 60 neurons. The solutions approximated by the Neurodifeq approach were compared with the exact solutions. The different results obtained are illustrated using various figures and tables.

4.1. Problem 1

When $p = 2$, then Newell-Whitehead-Segel's equation is as (33)

$$\frac{\partial v}{\partial t} = \beta \frac{\partial^2 v}{\partial x^2} + cv + kv^2, \quad x \in [0, 1], \quad t \in [0, 1] \quad (33)$$

with the initial condition as in (34):

$$v(x, 0) = -\frac{3\lambda^2\beta}{2k} \left(1 - 2 \tanh\left(\frac{\lambda x}{2}\right) + \tanh^2\left(\frac{\lambda x}{2}\right) \right) \quad (34)$$

the boundary conditions as in (35)

$$\begin{aligned} v(0, t) &= -\frac{3\lambda^2\beta}{2k} \left(1 - 2 \tanh\left(\frac{\lambda}{2}(-at)\right) + \tanh^2\left(\frac{\lambda}{2}(-at)\right) \right), \\ v(1, t) &= -\frac{3\lambda^2\beta}{2k} \left(1 - 2 \tanh\left(\frac{\lambda}{2}(1-at)\right) + \tanh^2\left(\frac{\lambda}{2}(1-at)\right) \right) \end{aligned} \quad (35)$$

and the exact solution as in (36):

$$v_{ex}(x, t) = -\frac{3\lambda^2\beta}{2k} \left(1 - 2 \tanh\left(\frac{\lambda}{2}(x-at)\right) + \tanh^2\left(\frac{\lambda}{2}(x-at)\right) \right) \quad (36)$$

with $c = 6\lambda^2\beta$, $a = 5\lambda\beta$, $k = 0.5$, $\beta = 0.2$, and $\lambda = 0.8$ which represent real-valued constants, we evaluated the effectiveness of the NeuroDiffEq approach in solving the second differential by (33). Figure 2 illustrates the comparison between the exact solution and the approximate solution obtained using an ANN. The figure presents both solutions: the curve on the right corresponds to the exact solution, while the curve on the left represents the ANN-based approximation.

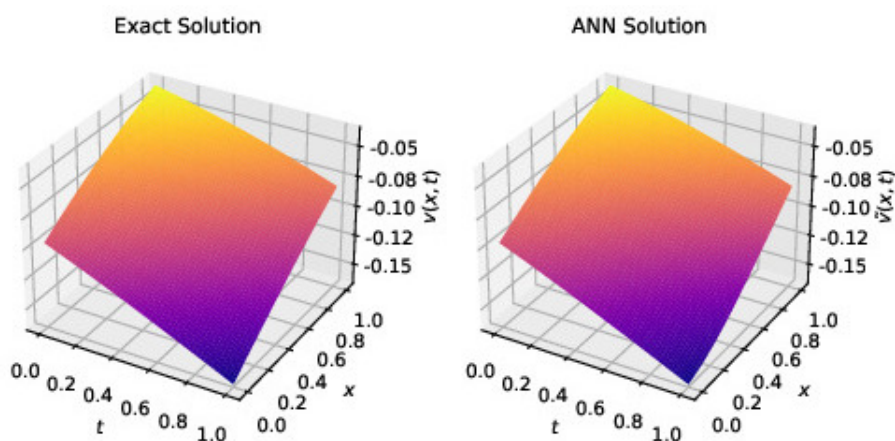


Figure 2. The exact and the ANN-solution of (33)

We can observe that the two curves overlap almost perfectly, indicating that our deep learning-based model is capable of accurately reproducing the behavior of the differential equation. This visual comparison highlights the effectiveness of the ANN in solving the second differential equation. The approximate solution aligns precisely with the exact one, confirming the model's ability to capture the fundamental dynamics of the problem.

We also assessed the model's performance using Table 1, which reports the loss values for different numbers of network layers. As in the previous case, we observed that the loss decreases as the number of hidden layers increases, thereby emphasizing the improved accuracy and robustness of the model with increased architectural complexity.

Table 1. Values of the loss of (33) at various layers numbers

Hidden layers	3-layers			4-layers			5-layers		
Neurons	20	30	40	20	30	40	50	60	
Errors	1.9×10^{-5}			9.4×10^{-6}			9×10^{-6}		

4.2. Problem 2

When $p = 3$, then Newell-Whitehead-Segel's equation is as (37):

$$\frac{\partial v}{\partial t} = \beta \frac{\partial^2 v}{\partial x^2} + cv + kv^3, \quad x \in [0, 1], \quad t \in [0, 1] \quad (37)$$

with the initial condition as in (38):

$$v(x, 0) = -\lambda \sqrt{\frac{\beta}{2k}} \left(-1 + \tanh\left(\frac{\lambda x}{2}\right) \right) \quad (38)$$

and the boundary conditions as in (39):

$$v(0, t) = -\lambda \sqrt{\frac{\beta}{2k}} \left(-1 + \tanh\left(\frac{\lambda}{2}(-at)\right) \right), \quad v(1, t) = -\lambda \sqrt{\frac{\beta}{2k}} \left(-1 + \tanh\left(\frac{\lambda}{2}(1 - at)\right) \right) \quad (39)$$

with $c = 2\lambda^2\beta$, $k = 0.5$, $\beta = -0.2$, and $\lambda = 0.8$ represent the real numbers and $a = 3\lambda\beta$ represents velocity. The exact solution is as depicted in (40):

$$v_{ex}(x, t) = -\lambda \sqrt{\frac{\beta}{2k}} \left(-1 + \tanh\left(\frac{\lambda}{2}(x - at)\right) \right). \quad (40)$$

We have compared the exact solution with the approximate solution obtained using an ANN to evaluate the effectiveness of our deep learning-based approach to solving the third differential (37). Figure 3 illustrates this comparison using 3D visualizations. The figure on the left shows the exact solution, while the figure on the right displays the approximate solution obtained using the ANN.

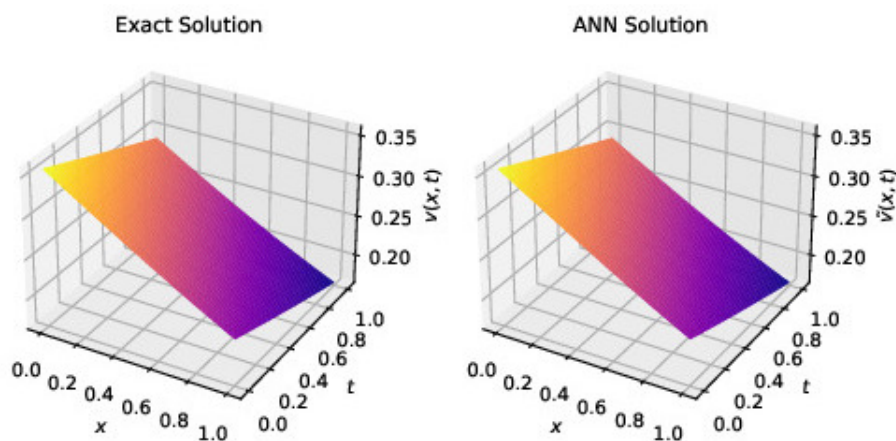


Figure 3. The exact and the ANN-solution of (37)

It can be seen that the two surfaces overlap closely, indicating that our deep learning-based model is capable of accurately reproducing the behavior of the differential equation. This 3D visual comparison

between the two solutions confirms the success of our method in solving the third differential equation. The approximate solution aligns very well with the exact solution, demonstrating the model's ability to capture the essential features of the problem.

We also evaluated the effectiveness of our model using Table 2, which presents the loss values for different numbers of network layers. As mentioned above, we observe that increasing the number of layers leads to a decrease in loss. This result underlines the improved performance of the model with higher architectural complexity.

Table 2. Values of the loss of (37) at various layers numbers

Hidden layers	3-layers			4-layers				5-layers				
Neurons	20	30	40	20	30	40	50	20	30	40	50	60
Errors	1.8×10^{-5}			5×10^{-6}				3×10^{-6}				

5. CONCLUSION

In this article, we examined the general Newell-Whitehead-Segel equations both numerically and analytically by applying the NeuroDiffEq and Unified approaches, respectively. The approximate solutions obtained were compared with the exact analytical solutions, showing that the NeuroDiffEq approach provides reliable and efficient approximations with minimal manual intervention. The results demonstrate a strong agreement with the exact solutions, confirming that deep learning particularly ANNs is a promising tool for solving differential equations, including both linear and nonlinear cases in applied sciences. However, this study also presents some limitations. The accuracy and convergence of the neural network models can be sensitive to hyperparameter choices and the selection of training data points. Furthermore, the computational cost may increase significantly for complex or high-dimensional problems. Future research should explore adaptive training techniques, automated hyperparameter optimization, and the application of these models to more complex boundary conditions or irregular geometries. Integrating physics-informed architectures with domain knowledge could also further improve both efficiency and generalization.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Soumaya Nouna	✓	✓	✓	✓	✓	✓		✓	✓		✓		✓	
Ilyas Tammouch	✓		✓	✓	✓	✓			✓		✓		✓	
Assia Nouna		✓	✓		✓		✓		✓	✓	✓	✓		
Mohamed Mansouri		✓	✓		✓		✓			✓				

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal Analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project Administration

Fu : Funding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Authors state no conflict of interest.




DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article. No additional datasets were generated or analyzed during the current study.




REFERENCES

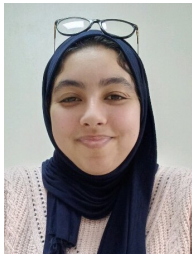
- [1] N. S. Elgazery, "A periodic solution of the Newell-Whitehead-Segel (NWS) wave equation via fractional calculus," *Journal of Applied and Computational Mechanics*, vol. 6, pp. 1293–1300, 2020, doi: 10.22055/JACM.2020.33778.2285.
- [2] N. H. Tuan, R. M. Ganji, and H. Jafari, "A numerical study of fractional rheological models and fractional Newell-Whitehead-Segel equation with non-local and non-singular kernel," *Chinese Journal of Physics*, vol. 68, pp. 308–320, 2020, doi: 10.1016/j.cjph.2020.08.019.
- [3] P. P-On, "Laplace adomian decomposition method for solving newell-whitehead-segel equation," *Applied Mathematical Sciences*, vol. 7, no. 129–132, pp. 6593–6600, 2013, doi: 10.12988/ams.2013.310603.
- [4] J. Singh and D. Kumar, "Homotopy perturbation algorithm using laplace transform for gas dynamics equation," *Journal of Applied Mathematics, Statistics and Informatics*, vol. 8, no. 1, pp. 55–61, 2012, doi: 10.2478/v10294-012-0006-2.
- [5] A. Prakash and M. Kumar, "He's variational iteration method for the solution of nonlinear newell-whitehead-segel equation," *Journal of Applied Analysis and Computation*, vol. 6, no. 3, pp. 738–748, 2016, doi: 10.11948/2016048.
- [6] J. E. M-Díaz and J. R-Ramírez, "A non-standard symmetry-preserving method to compute bounded solutions of a generalized newell-whitehead-segel equation," *Applied Numerical Mathematics*, vol. 61, no. 4, pp. 630–640, 2011, doi: 10.1016/j.apnum.2010.12.008.
- [7] N. Wandel, M. Weinmann, and R. Klein, "Learning incompressible fluid dynamics from scratch - towards fast, differentiable fluid models that generalize," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.
- [8] A. Kashefi, D. Rempe, and L. J. Guibas, "A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries," *Physics of Fluids*, vol. 33, no. 2, 2021, doi: 10.1063/5.0033376.
- [9] A. Malek and R. S. Beidokhti, "Numerical solution for high order differential equations using a hybrid neural network–optimization method," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 260–271, 2006, doi: 10.1016/j.amc.2006.05.068.
- [10] J. Sirignano and K. Spiliopoulos, "DGM: a deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018, doi: 10.1016/j.jcp.2018.08.029.
- [11] S. Nouna, A. Nouna, M. Mansouri, I. Tammouch, and B. Achchab, "Two-dimensional Klein-Gordon and Sine-Gordon numerical solutions based on deep neural network," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 2, pp. 1548–1560, 2025, doi: 10.11591/ijai.v14.i2.pp1548-1560.
- [12] S. Nouna, A. Nouna, M. Mansouri, and A. Boujamaa, "Utilizing deep learning algorithms for the resolution of partial differential equations," *International Journal of Informatics and Communication Technology*, vol. 13, no. 3, pp. 370–379, 2024, doi: 10.11591/ijict.v13i3.pp370-379.
- [13] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.
- [14] Y. Guo, X. Cao, B. Liu, and M. Gao, "Solving partial differential equations using deep learning and physical constraints," *Applied Sciences*, vol. 10, no. 17, 2020, doi: 10.3390/app10175917.
- [15] A. Paszke *et al.*, "Automatic differentiation in PyTorch," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, CA, USA, 2017.
- [16] M. Abadi *et al.*, "Tensorflow: a system for large-scale machine learning," in *OSDI'16: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pp. 265–283, 2016.
- [17] S. Panghal and M. Kumar, "Optimization free neural network approach for solving ordinary and partial differential equations," *Engineering with Computers*, vol. 37, no. 4, pp. 2989–3002, 2021, doi: 10.1007/s00366-020-00985-1.
- [18] F. Chen *et al.*, "NeuroDiffEq: a python package for solving differential equations with neural networks," *Journal of Open Source Software*, vol. 5, no. 46, 2020, doi: 10.21105/joss.01931.
- [19] H. I. A.-Gawad, "Towards a unified method for exact solutions of evolution equations. An application to reaction diffusion equations with finite memory transport," *Journal of Statistical Physics*, vol. 147, no. 3, pp. 506–518, 2012, doi: 10.1007/s10955-012-0467-0.
- [20] H. I. A.-Gawad and M. Osman, "Exact solutions of the Korteweg-de Vries equation with space and time dependent coefficients by the extended unified method," *Indian Journal of Pure and Applied Mathematics*, vol. 45, no. 1, pp. 1–12, 2014, doi: 10.1007/s13226-014-0047-x.
- [21] H. I. A.-Gawad, M. Osman, and N. S. Elazab, "Exact solutions of space-time dependent Korteweg-de Vries equation by the extended unified method," *Life Science Journal*, vol. 10, no. 2, pp. 2598–2604, 2013, doi: 10.9790/5728-0460106.
- [22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, doi: 10.1016/0893-6080(89)90020-8.
- [23] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, IEEE, 2018, pp. 1836–1841, doi: 10.1109/CCDC.2018.8407425.
- [24] K. Shukla, A. D. Jagtap, and G. E. Karniadakis, "Parallel physics-informed neural networks via domain decomposition," *Journal of Computational Physics*, vol. 447, 2021, doi: 10.1016/j.jcp.2021.110683.
- [25] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, doi: 10.1038/s42254-021-00314-5, 2021.




BIOGRAPHIES OF AUTHORS

Soumaya Nouna    is a researcher at the Systems Analysis and Decision Support Laboratory at Hassan First University in Settat, Morocco. She holds a Ph.D. in applied mathematics and computer science. Her expertise spans applied mathematics, machine learning, and deep learning, with a particular focus on the analysis and numerical solution of differential equations using AI-based methods. She has authored several scientific papers published in international journals and conferences. Her research bridges advanced mathematical theory with artificial intelligence to address real-world challenges. She can be contacted at email: s.nouna@uhp.ac.ma.






Dr. Ilyas Tammouch    is a dedicated researcher and academic affiliated with Ibn Tofail University in Kenitra, Morocco. His work covers a wide range of topics in artificial intelligence, including machine learning, deep learning, data analytics, and performance evaluation systems. He has made significant contributions to research, focusing on practical, data-driven solutions across sectors such as education, healthcare, and industry. Beyond the technical dimension, his research also addresses the ethical and societal implications of emerging technologies. He actively participates in national and international academic collaborations and knowledge-sharing initiatives. He can be contacted at email: ilyas.tammouch@uit.ac.ma.



Assia Nouna    is a doctoral researcher at the Systems Analysis, Modelling and Decision Support Research Laboratory at Hassan First University in Settat, Morocco. She is currently pursuing a Ph.D. in mathematics and computer science. Her research focuses on deep learning and satellite imagery for agricultural applications, aiming to improve farming practices through precise soil analysis, better crop management, and more accurate yield predictions. In addition to her academic work, she has contributed to several projects and publications, demonstrating her ability to apply advanced computational methods to real-world challenges. She can be contacted at email: a.nouna@uhp.ac.ma.



Mohamed Mansouri    received his Ph.D. in Mechanical Engineering and Engineering Sciences in 2013 through a joint program between the Faculty of Science and Technology at Hassan First University in Settat, Morocco, and INSA Rouen, France. He is currently a Professor and researcher at the National School of Applied Sciences (ENSA) in Berrechid, within the Department of Electrical Engineering and Renewable Energies. His research interests include mechano-reliability analysis, industrial engineering, shape and reliability optimization of coupled fluid–structure systems, as well as energy storage technologies. He can be contacted at email: mohamed.mansouri@uhp.ac.ma.