

PDE Modelling of Population Dynamics using Physics Informed Neural Networks (PINNs)

Anthony Udo Akpan

Department of General Studies,
Akwa Ibom State Polytechnic, Ikot Osurua, Ikot Ekpene
tonny.akpan@gmail.com

Abstract

In recent years, Artificial Intelligence (AI) has transformed scientific inquiry and technological innovation significantly in almost all facets of life. This paper presents a study of Physics-Informed Neural Networks (PINNs) solution techniques applied to Partial Differential Equation (PDE) models in population dynamics. Specifically, the paper focuses on modelling disease spread using an advection-diffusion-reaction partial differential equations (PDEs), with the solution sought through Physics-Informed Neural Networks (PINNs) technique. The community is being modelled as a bounded spatial domain where the disease density evolves over time and space. By embedding the underlying physical and biological laws into the network architecture, PINNs offer a robust and accurate framework to simulate infectious disease dynamics. Furthermore, numerical simulations, were implemented using the MATLAB ODE45 scheme, which provided insights into the interplay between disease progression, recovery, birth and death rate as parameter of interest in the transmission dynamics.

Keywords: PDE modelling, population dynamics, physics informed neural network, machine learning, biological laws, disease spread.

Introduction

Population dynamics refer to study of the variation in the size and density of population over time and space reflecting the net effect in differences among individuals in their physiological and behavioural interactions with the environment. It represents the changes in the number a species in a single location. Population models have been germane and the time dependent interactions between modelling species have been of great interest to ecologists over the years.

Beginning from the well-known Lokta-Volterra predator prey equations derived in the 1920s, mathematical modellers have utilised these equations to describe even much more complex systems in more biological setting such as competition, symbiosis, disease model (SIR -Susceptible Infected-Recovered typed model) powered mostly by ordinary differential equations and a host of others. However, as noted by Christou, (2022) this set of ordinary differential equations fails to capture the spatial effect and thus, the inclusion of diffusion terms and spatial dependence in the work by Conway and Smoller (2007), birthed the use of PDE partial differential equations (PDEs) in modelling dynamics.

Partial differential equations (PDEs) are pivotal to the modelling of natural phenomena and find applications in almost every field of science and engineering driving the means to tackling vast and ever-expanding array of real-world problems—from the simple heat equation to more complex systems describing financial markets, weather forecasting, or population dynamics. In the context of disease spread, PDEs serve as a powerful technique to capture the dynamics of infectious agents over space and time. The desire to understand the solutions to these equations has over the years pre- occupied mathematicians, scientist as these solutions offer both insights into the underlying phenomena and valuable predictive capabilities.

Traditional numerical techniques for approximating PDEs, such as finite difference and finite element methods, have been thoroughly developed and refined over the years. These methods typically involve discretising the spatial domain using a mesh, which transforms the PDE into a system of ordinary differential equations (ODEs) that can be time-stepped to approximate the original problem. Though highly robust and reliable, these grid-based methods require copiously fine discretisations to enhance accuracy. For stability considerations, finer spatial discretisations requires smaller time steps, resulting in more computationally expensive schemes.

Neural network describes a mathematical convenient and simplified version of neurons in a brain encapsulating elements called 'perceptrons'. Neural network is made up of a large network of these perceptrons just as the brain is a big network of neurons.

In recent years, advances in Artificial Intelligence (AI) have opened new vistas for solving PDEs. Solving partial differential equations (PDEs) governing physical phenomena using machine learning (ML) has emerged as a new field of scientific machine learning leveraging the universal approximation theorem and high expressivity of neural networks. Physics-Informed Neural Networks (PINNs) represent a groundbreaking technique that roots the physical laws—expressed as PDEs—directly into the neural network's training process. Rather than relying wholly on discretisation, PINNs enforce the consistency with observed data and the underlying PDE constraints simultaneously by incorporating both into a composite loss function. For modelling disease spread, this methodology is particularly attractive, since it enables the integration of sparse or noisy observational data with the rigorous mathematical structure provided by an advection–diffusion–reaction PDE.

The recent forays of scientists and modellers into physics informed neural network (PINN) otherwise known as theory trained neural network (TTNN) to provide solutions to PDE systems is well documented. For example, Raissi et al., (2019) introduced a novel framework that leverages machine learning (ML) integrated with physics-based constraints both to solve and understand partial differential equations (PDEs) from data. Their approach comprises two complementary strategies: a continuous time model that embeds PDE residuals into the loss function to create data-efficient spatio-temporal approximators, and a discrete time model that employs implicit Runge–Kutta schemes for highly accurate temporal integration. This dual technique has been effectively utilised across diverse spectrums—including fluid dynamics, quantum mechanics, reaction–diffusion systems, and nonlinear shallow-water wave propagation—demonstrating its capacity to not only simulate complex systems with limited data but also uncover underlying physical laws (Wu et al., 2024).

For Rodrigues, (2024), the main idea behind PINNs is to train neural networks to learn not only from observed data but also adhere to the underlying physics that govern the system. This is achieved by incorporating differential equations or other relevant physical constraints as additional terms in the loss function during training. This unique combination allows PINNs to generalise well beyond the available data and offers a data-driven framework for solving complex physical problems.

Lin and Chen (2023) proposed a twin novel physics-informed neural network (PINN) schemes that integrate Miura transformation constraints into the learning process to solve nonlinear partial differential equations (PDEs) using an unsupervised approach. Their method leverages the Miura transformation as a critical bridge, allowing initial-boundary data from one nonlinear equation to drive the data-driven solution of another. Through extensive computational experiments

on the KdV and mKdV equations, the authors not only reproduced the dynamic behaviour of the solutions but also discovered a new localised wave—namely, a kink-bell type solution for the defocusing mKdV equation. Their comparative analysis of the two schemes underscores that each has its own merits, suggesting that the choice of method should be tailored to the specific problem at hand. The utilisation of PINNs by several authors Rodrigues ((2024); Lin and Chen (2024); Wu et al. (2024) to study different phenomena directed by PDE systems underscore the growing influence of Artificial Intelligence (AI) enabled solutions in everyday life.

In this study, a novel PDE modelling technique of population dynamics specifically disease transmission dynamics using the Physics informed neural networks (PINNs) solution techniques is formulated and analysed qualitatively and quantitatively. The study is further structured for ease of presentation as follows: Section one deals with the introduction while section two entertains the mathematical formulations and assumptions. Section three is engulfed in the analysis involving both qualitative and quantitative approach. Section four is concerned with the simulations and discussions of the results. The paper concludes in section five with further research direction.

Mathematical Formulations and Assumptions

Disease spread in a typical population amongst species is characterised by a complex interaction such as spatial movement representing the advection terms, random movement (diffusion) and local interactions (reactions) such that the governing equation is directed by a PDE. In this study, an advection-reaction diffusion model describes the spread of a disease in a local population and is presented thus:

$$\frac{\partial u}{\partial t} + \nabla \cdot (v u) = D \nabla^2 u + R(u, x, t) \quad (2.1),$$

Where $u(x, t)$ is the disease density at location x and time t .

v is the advection velocity field representing the human movement

D is the diffusion coefficient and $R(u, x, t)$ is the reaction terms modelling the local infection dynamics such as infectivity, recovery and death with $R(u, x, t) = \beta u(1 - u) - uy$ and follows a logistic format, β being the infection transmission rate and y is the infection recovery rate. A feed forward neural network was designed to approximate the solution of system (2.1). The network architecture included an input layer, multiple hidden layers, and an output layer corresponding to the predicted population size. The loss function is modelled below:

$$\mathcal{L}(\varphi) = \mathcal{L}_{data} + \lambda \mathcal{L}_{PDE} \quad (2.2)$$

Where $\mathcal{L}_{data} = \frac{1}{N_d} \sum_{i=1}^{N_d} |U(x_i; t_i; \theta) - u_i|^2$ is the measure of error with respect to the observed disease data and λ is the regularisation parameter.

$\mathcal{L}_{PDE} = \frac{1}{N_r} \sum_{r=1}^{N_r} \left| \frac{\partial U}{\partial t} + \nabla \cdot (v U(x_j; t_j; \theta)) - D \nabla^2 U(x_j; t_j; \theta) - R(U(x_j; t_j; \theta), x, t) \right|^2$ is the measure of the PDE solutions ensuring that the governing equation is enforced in this instance by penalising any violation of the underlying physical laws at the point of selection in the domain of consideration.

PINN Architecture

The algorithm for the implementation of PINN is outlined in these sections. This is achieved as follows:

1. Define the PDE parameters by setting values for Advection, diffusion coefficient, infection transmission rate and recovery rate respectively v, D, β and γ .
2. Define the normalised domain as $\Omega := x \in [0, 1]$ and $t \in [0, 1]$ for the spatial location and time.
3. The N_{data} point along the spatial domain at $t = 0$ is generated using the Gaussian profile centred at $x = 0.5$ to simulate the initial outbreak: $u(x, 0) = \exp(-100(x - 0.5)^2)$
4. The N_{colloc} random points $x(x, t)$ is generated such that the enforced neural network output satisfies the governing PDE throughout the domain.
5. The neural network setup is implemented manually using the feed-forward neural network with two input layers corresponding the x and t and two hidden layers with 20 neurons using the hyperbolic tangent (\tanh) activation function and a single neuron output that predicted the disease density $u(x, t)$.
- 6) All network parameters (weights and biases) are initialised using small random values and stored in a single vector θ .
- 7) The loss function is defined by computing the mean square error (MSE) between the predicted neural network and the initial observed data (observed error) such that:

$$\mathcal{L}_{Loss_{data}} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} u_{NN}(x_i; 0; \theta) - u_{data}(x_i))^2$$

- 8) Define the PDE loss (residual error) by approximating the derivatives of $u(\vec{x}, t)$ using central finite differences such that

the time derivative $u_t = \frac{u(x, t+\epsilon) - u(x, t-\epsilon)}{2\epsilon}$ and the spatial derivative

$u_x = \frac{u(x+\epsilon,t) - u(x-\epsilon,t)}{2\epsilon}$ and the second spatial derivative $u_{xx} = \frac{u(x+\epsilon,t) - 2u(x,t) + u(x-\epsilon,t)}{\epsilon^2}$. Forming the PDE residual function $f = u_t + vu_x - Du_{xx} - \beta u(1 - u) - \gamma u$ and computing the PDE loss as the MSE of this residual such that

$$\text{Loss}_{PDE} = \frac{1}{N_{colloc}} \sum_{i=1}^{N_{colloc}} f(x_j, t_j)^2$$

9) The combined data and PDE loss function is calculated:

$$\text{Loss}_{total} = \text{Loss}_{PDE} + \text{Loss}_{data}$$

10) Define the objective function that returned the total loss function given by the parameter θ and thereafter utilised the trained network to predict $u(x, t)$ over the grid covering the entire domain $u(x, \mathbb{D})$.

Based on the above algorithm generated using MATLAB is as shown in the table below:

Table 1

Algorithm generated using MATLAB

Iteration	Func-count	f(x)	Step-size	optimality
0	502	0.186273		0.781
1	1506	0.13376	0.321166	0.307
2	2008	0.109949	1	0.123
3	2510	0.104914	1	0.0568
4	3012	0.103023	1	0.0249
5	3514	0.096667	1	0.0776
6	4016	0.094622	1	0.0426
7	4518	0.0939039	1	0.00619

8	5020	0.093862	1	0.00111
9	5522	0.0938617	1	0.00131
10	6024	0.0938607	1	0.00153
11	7028	0.0938599	0.377604	0.00179
12	8032	0.0938595	0.282073	0.00144
13	13554	0.0938591	0.346832	0.00154
14	14056	0.0938567	1	0.00131
15	14558	0.0938553	1	0.00125
16	15060	0.0938551	1	0.00139
17	15562	0.0938519	1	0.00132
18	16064	0.0938492	1	0.00214
19	16566	0.0938439	1	0.00385
20	17068	0.0938292	1	0.00405

Numerical Simulations

As noted by Chin & Mckay, (2019) model verification and validation can provide evidence that a given model is accurate and hence, simulation of the model system (2.1) was carried out with the data to represent and understand the research problem. Since it is based on dearth of mathematical modelling on the subject, all values are assumed for simulation purposes.

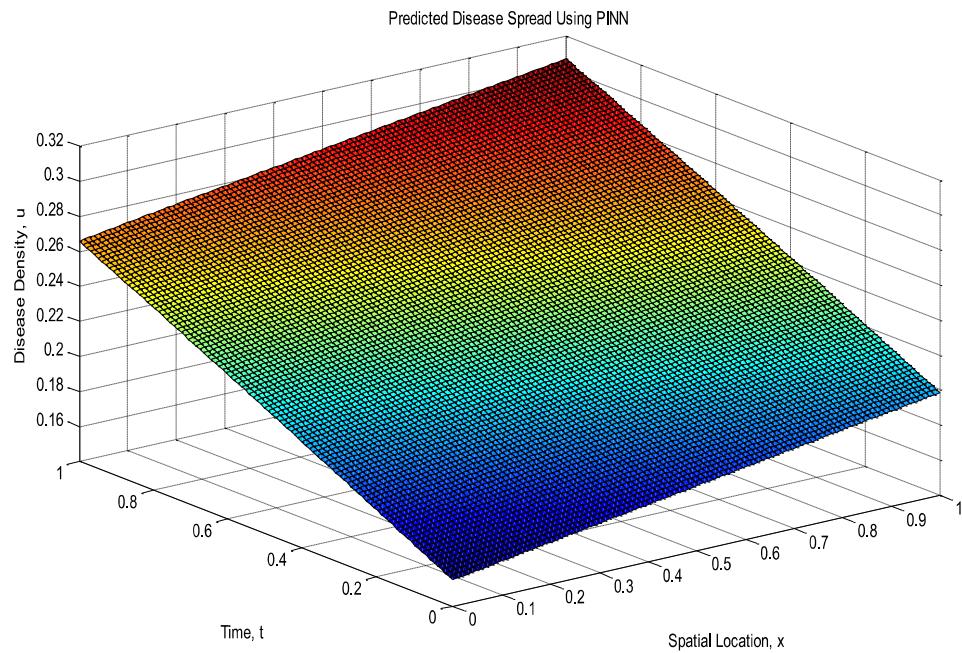


Figure 1: A 3D plot in a normalised spatial domain $\Omega: x \in [0, 1]$ of disease spread in a typical community where the simulations begin at $t = 0$ and $t = 1$ capturing how the disease propagates over time. u represent the prevalence of the disease at each spatial location x and time t .

The figure depicts a gradual increase in the density of the disease from blue (low points) to red (high points) as the time progresses. As could be seen from the figure, there are no sharp peaks or localised hotspot evidently suggesting a uniform propagation over time and space rather than a sudden outbreak. The sloped surface shows a steady increase in disease density across space and time in the domain of interest suggesting that the governing equations enforce a smooth continual spread in the local community. The absence of sharp variations along the x – axis is indicative of a diffusion dominated spread leading to a uniform distribution across the local community of interest. The implication of the gradual spread of the disease is suggestive that diffusion is the primary driver rather than advection. This might be akin to a scenario where the recovery or removal terms are not enforced mirroring the early stages of an epidemic where infections dominate over recovery.

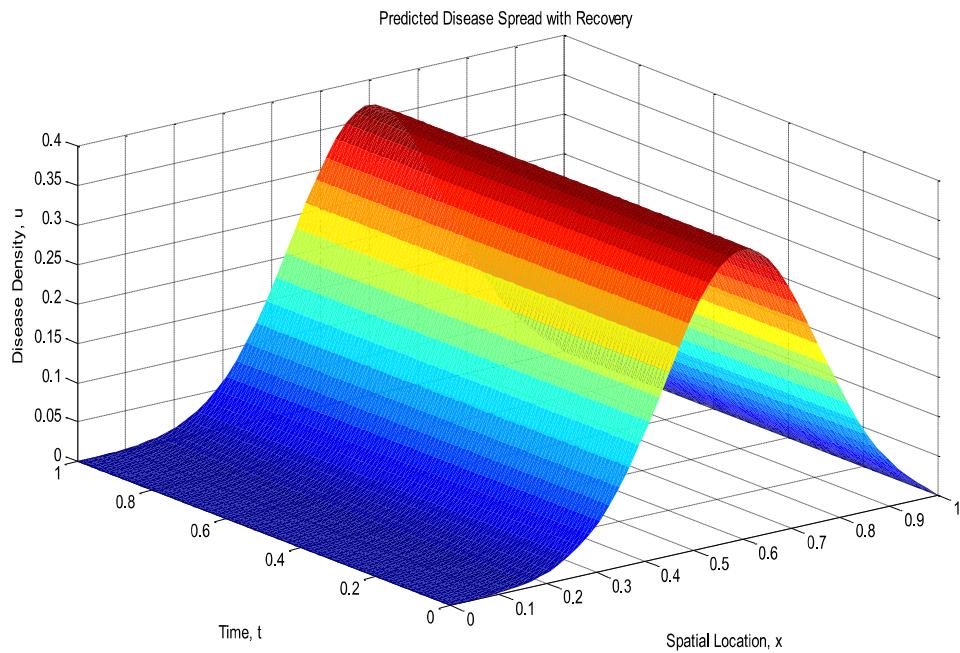


Figure 2: A 3D plot of disease spread $u(x, t)$ over time t and location x in a typical community over a spatiotemporal domain incorporating diffusion, advection and recovery. The plot showcases the fact that at time $t = 0$ the infection begins as a localised peak in the middle of the region of the local community with the initial peak representing high infection concentration at the specific spatial location. Over time, the disease spread outward, modelling the random movement (diffusion) of infected individual while the rightward shift shows disease spread in a preferred direction (advection). The infectivity peaked at the red (high point) and as individual recovery ensued leading to reduction in the infective density and as this happens over time then fewer individuals remain infected. Thus, as $t \rightarrow 1$ the infective individual reduces significantly showing the triple effects of diffusion, advection and recovery. Though the infection is not completely eliminated but it is diminished significantly. This is an indicative tendency towards disease control that could be achieved in this instance. The result underpins that disease diffusion, transport and recovery interact dynamically.

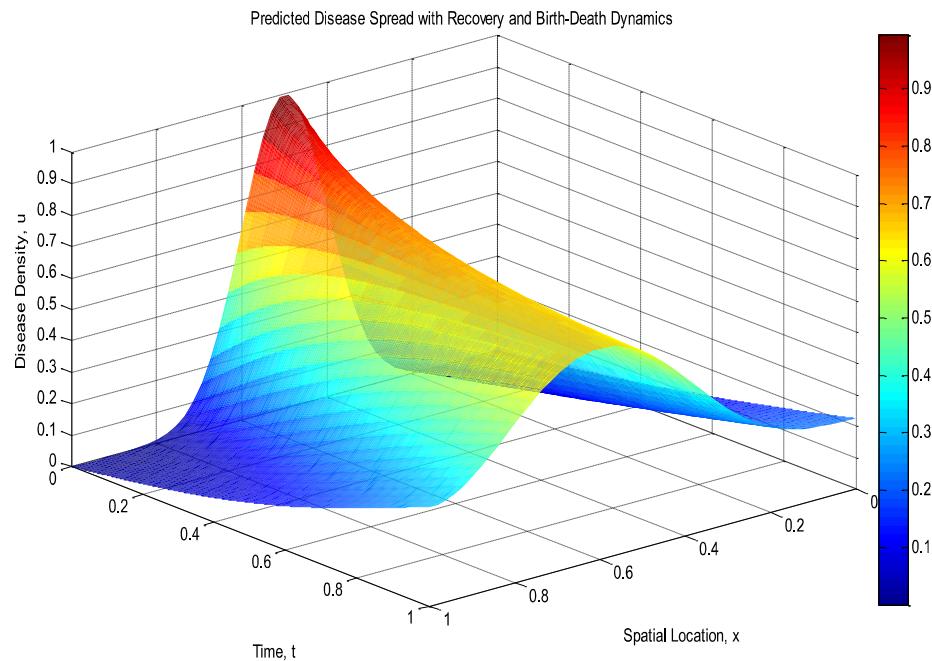


Figure 3: The plot illustrates the propagation of disease in a typical community where birth and death are taken into account.

Discussion of Results

The simulations above as depicted in the visualisation seen in figures (1-3) showcase a simple disease propagation under several modelling scenarios ranging from basic disease transmission, recovery mechanism and birth-death dynamics proving key insights that could be utilised for quick decision-making process in a public health setting.

The initial plot mirrors a monotonically increasing surface indicating that disease density increases over time and spatial location. The absence of the recovery mechanism implies that once an infective individual remains infected in a population leading to a gradual accumulation of the disease in the typical human society could lead to the prevalence of the disease in such a situation. Expectedly, in a purely diffusion-based model, where infections propagate freely without any recovery; the disease continues to spread unabated until it saturates the entire population - a reflection of unchecked epidemic.

On the other hand, the second plot (figure 2) reflect a non-monotonic trend where initially the disease increases, peaked and finally decline over time, analogous to the fact that the infected individual recover over time, resulting the decline in disease

prevalence over the spatial domain. The inclusion of the recovery mechanism significantly changes disease progression. The maximum burden of the disease is represented at the peak and thereafter recovered individual lowered the disease density, akin to a typical real-world epidemic scenario where disease outbreaks have peaked phase followed by declining phase possibly due to immunity or interventions.

Finally, the third plot represents a complex scenario owing to the introduction of birth-death into the dynamics. In this instance new susceptible individual are introduced by birth whereas infected individual could die reducing disease density. Thus, the disease does not necessarily vanish out of the population but reaches steady states depending on the birth, infection, recovery and death rate. The implication of this situation is that if the birth rate exceeds the death rate, then the new susceptible will enter the population enabling the disease to persist indefinitely in the population. In the same vein, if the recovery and death interplay is high enough, then the possibility of the disease to die out in the population is guaranteed. The model underscores a real-world infectious disease system where the population is dynamic, rather than being static.

Conclusion

This study has demonstrated the successful application of Physics-Informed Neural Networks (PINNs) to model disease spread through an advection-diffusion-reaction PDE. The approach effectively integrates sparse observational data with the underlying physical laws governing disease dynamics, overcoming several limitations of traditional numerical methods. By embedding the PDE constraints directly into the training process, PINNs provide a robust and computationally efficient means of forecasting epidemic evolution.

To illustrate the potential of this methodology, it was vital to simulate several scenarios of disease spread in a typical local community. Numerical simulations were implemented using the MATLAB ODE45 scheme, which provided insights into the relationship disease propagation, recovery, birth and death rate as parameter of interest in the transmission dynamics. The community is modelled as a bounded spatial domain where the disease density evolves. An outbreak is assumed to start from a localised region, figures (4.1-4.3) representing an initial cluster of infection within the community. The velocity field is configured to reflect common commuting patterns and daily movements, which are significant in shaping the directional spread of the disease.

Diffusion models the random movement of individuals across the community, accounting for local interactions that are not captured by directed movement. The reaction term modelled to follow a logistic growth framework captures the intrinsic

biological dynamics of disease transmission. The application of the PINNs underpins the utility of the method in solving PDE systems.

Recommendations

Based on the results of the study, the following recommendations are made:

- The model should be applied to real-world epidemiological data to demonstrate the practical relevance of PINNs instead of the current simulations which uses hypothetical community parameters.
- Integrating PINN with real-time data stream and online learning could enable dynamic predictions of outbreaks. This would provide timely insight for public health officials, thereby, improving response efficiency during emerging epidemic situations.
- Combining the PINNs with traditional finite element or difference numerical scheme to explore their synergy could be utilised for enhanced numerical accuracy and stability. This approach can leverage the interpretability of the classical method while exploiting the flexibilities and learning capacities of neural networks.

Suggestion for further Research

Further research avenues could explore the application of artificial intelligence for policy analysis and forecasting. This could involve utilising AI techniques to model and predict the impact of various policy interventions, including strategies to mitigate disease outbreaks.

References

Chin, C. & Mckay, A. (2019). Model verification and validation strategies: An application case study. *The 8th International Symposium on Computational Intelligence and Industrial Applications*. (ISCHA2018).

Christou, K. (2022). *Application of partial differential equations in population dynamics: Fixed-mesh and adaptive mesh approaches* (Doctoral dissertation, University of Reading).

Conway, E. D., & Smoller, J. A. (2007). A comparison technique for systems of reaction-diffusion equations. *Communications in Partial Differential Equations*, 32(10), 1531–1550. <https://doi.org/10.1080/03605300701318927>

Jagtap, A. D., Kawaguchi, K., & Karniadakis, G. E. (2020). Locally adaptive activation functions with slope recovery for deep and physics-informed neural

networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2239), 20200334.
<https://doi.org/10.1098/rspa.2020.0334>

Lalić, B., Cuong, D. V., Petrić, M., Pavlović, V., Sremac, A. F., & Roantree, M. (2024). Physics-based dynamic models hybridisation using physics-informed neural networks. *arXiv preprint arXiv:2412.07514*

Lin, S., & Chen, Y. (2023). Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions. *Physica D: Nonlinear Phenomena*, 445, 133629.
<https://doi.org/10.1016/j.physd.2022.133629>

Lin, S., & Chen, Y. (2024). Gradient-enhanced physics-informed neural networks based on transfer learning for inverse problems of the variable coefficient differential equations. *Physica D: Nonlinear Phenomena*, 459, 134023.
<https://doi.org/10.1016/j.physd.2023.134023>

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>

Rodrigues, J. A. (2024). *Using physics-informed neural networks (PINNs) for tumor cells growth modeling* [Preprint]. Preprints.org.
<https://doi.org/10.20944/preprints202403.1013.v1>

Wu, Y., Sicard, B., & Gadsden, S. A. (2024). Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring. *Expert Systems with Applications*, 255, 124678.
<https://doi.org/10.1016/j.eswa.2024.124678>