Physics Informed Deep Learning Approach for Differential Equation

A. Mohanapriya[1], A. Abirami[2], Gopal Thangavel[3], Shenbagavalli[4]

[1]Assistant Professor , Department of Mathematics, Christ college of Science and Management, Malur, Karnataka, India, amohana20@gmail.com
[2]Assistant Professor, Department of Mathematics, Sona College of Technology, Salem, Tamil Nadu, India, abiramia@sonatech.ac.in
[3]Lecturer, Department of Mathematics, Preparatory Studies Centre, University of Technology and Applied Sciences-Nizwa, Oman, gopal.thangavel@utas.edu.om
[4]Assistant Professor, Department of Mathematics, St. Francis De Sales College, Bangalore, Karnataka, India, shenbamaths@gmail.com

This study presents a Physics-Informed Neural Network (PINN) approach for solving differential equations, providing a versatile and data-driven alternative to traditional methods. Focusing on the comparison between exact, analytic, and neural network solutions, we investigate the effectiveness of PINNs in capturing complex dynamics across diverse applications. The comparison is illustrated through carefully crafted graphs, highlighting the accuracy and efficiency of the PINN methodology. By eliminating the need for explicit analytical solutions, PINNs offer a flexible framework for addressing a wide range of differential equations, showcasing their potential to revolutionize problem-solving in various scientific and engineering domains. This research contributes valuable insights into the capabilities of PINNs and their role in advancing computational methodologies for differential equations.

Keywords: Neural Networks, Differential Equations, Laplace Transforms, Logistic Differential Equation.

1. Introduction

Differential equations play a crucial role in modelling a diverse range of physical [1], biological [2], and engineering phenomena [3,4]. Achieving an optimal balance between accuracy and computational efficiency in solving these equations has been a longstanding challenge. While traditional numerical methods and analytical techniques have been widely employed, they face difficulties when handling complex, nonlinear systems or problems lacking explicit analytical solutions. In recent years, the advent of Physics-Informed Neural Networks (PINNs) has introduced innovative approaches to address differential equations by integrating principles of physics with the learning capabilities of neural networks [5-7].

Neural networks (NN) have become integral in solving problems across various domains, as highlighted in Le Cun et al. [8], ranging from computer vision to language processing. Notably, machine learning approaches, particularly in scientific computing and differential equations, have gained prominence. Raissi et al. [9] introduced the concept of infusing physics knowledge into the learning process of neural networks, unlocking new possibilities for applications across diverse fields. One key application of neural networks is in supervised learning, where the objective is to establish a mapping function between input objects and their corresponding output values. This involves training the neural network using a dataset with input/output pairs, minimizing the error between predicted and true solutions through optimization processes employing gradient algorithms and automatic differentiation, as discussed in Baydin et al. [10].

When applied to differential equations, supervised learning entails finding a mapping function between physical input variables (e.g., position, time) and the solution of the equation. However, this approach has limitations, such as poor extrapolation beyond the training data range and the requirement for a substantial amount of training data to prevent convergence issues. To overcome these limitations, PINNs are introduced. PINNs enhance traditional neural networks by incorporating additional physics-based information, proving particularly valuable in simulating physical and engineering systems governed by differential equations. This involves evaluating the solution at collocation points to ensure that the estimated solution satisfies the equations. A new loss function, representing the physics, is introduced and combined with the original one during the learning process, leading to a dual-constraint approach that imposes restrictions in the solution space. This makes PINNs suitable for scenarios with limited known data, as often encountered in differential equations.

The use of PINNs for solving differential equations represents a departure from conventional methods, providing a data-driven and computationally efficient alternative. Unlike traditional approaches requiring explicit analytical solutions, PINNs leverage neural networks to directly approximate unknown solutions based on observed data and governing physical laws. This integration of machine learning techniques with the inherent physics of the problem offers a versatile tool for solving a wide range of differential equations, even in cases where traditional methods face challenges. This introduction sets the stage for exploring the application of PINNs in solving differential equations. As we delve into the intricacies of this methodology, we will examine how PINNs encode physical principles, learn from data, and provide accurate approximations of solutions.

The paper is organized as follows: Section 2 provides a review of the basics of PINNs for Ordinary Differential Equations (ODEs). Section 3 offers a tutorial on the method, demonstrating its application to solve simple linear and nonlinear differential equations. Finally, Section 4 draws conclusions based on the insights gained from the exploration of PINNs in solving differential equation.

2. Physics Informed Neural Network

The Basics of PINNs for ODE:

A Physics-Informed Neural Network (PINN) is a specialized artificial neural network designed to integrate established physical principles or equations into its architecture. By leveraging the governing equations within the framework of deep learning, a PINN capitalizes on domain-specific knowledge to enhance its ability to learn and predict outcomes. This unique approach allows the network to respect the underlying physics of the system it models, fostering improved accuracy and generalization in predicting physical phenomena. In essence, a PINN harmoniously combines the power of machine learning with domain-specific insights, contributing to more robust and accurate predictions in line with the governing physics of the system.

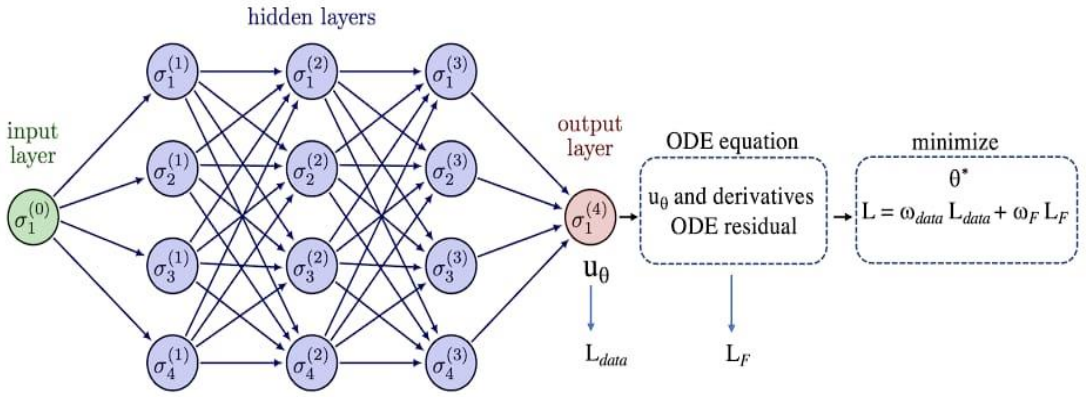

Figure 1: illustrates the schematic structure of a Physics-Informed Neural Network employed in solving a differential equation. The input layer comprises a single input variable, such as a time coordinate, represented by one neuron. The network consists of three hidden layers, each housing four neurons, interconnecting with both the input and output layers. The output layer, hosting a single variable (one neuron), symbolizes the solution denoted as u_θ.

3. Illustration of the Method on a Simple Examples

A solution in neural networks typically involves determining the optimal weights and biases that minimize a specific loss function. The objective is to fine-tune these weights and biases to reduce the overall loss and enhance the network's capability to provide accurate predictions for unseen data.

The Physics-Informed Neural Network (PINN) algorithm entails training a neural network to concurrently assimilate insights from available data and conform to established physics or mathematical principles. Here is a simplified overview of the process:

1. Data Collection: Collect relevant data pertaining to the problem at hand, which may consist of input-output pairs reflecting the system's behavior.

2. Neural Network Architecture: Devise a neural network architecture tailored to the specific

problem. Emphasize the inclusion of layers capable of learning from data while incorporating physics equations as constraints.

- Loss Function: Define a comprehensive loss function that encompasses two key components:

1. Data Loss Component 2. Regularization Component

- Data Fitting Loss: Evaluates the disparity between the neural network predictions and the actual data.

3. Physics-Informed Loss: Embeds the physics equations or principles as constraints, ensuring adherence to the underlying laws.

4. Training: Employ optimization techniques to minimize the combined loss function. This process trains the neural network to deliver accurate predictions while conforming to the specified physics constraints.

5. Validation: Evaluate the performance of the trained model on validation data to ascertain its generalization capabilities. This ensures that the model captures both the inherent data patterns and adheres to physics principles.

The distinctive feature of PINN lies in its capacity to harness data for learning while incorporating prior knowledge of the system's physics. This makes PINN particularly valuable for scenarios where data is limited, yet the underlying physics are well-understood.

I Linear Differential Equation

We have employed PINN to solve a range of linear differential equations with initial conditions. The table provides a list of the specific problems addressed. The solutions for each problem were obtained through Python implementation, and the corresponding visual representations are depicted in Figures 2 to 4.

S.No	Differential Equation	Initial Condition
1	$\frac{du}{dx} = x$	u(0)=0
2	$\frac{du}{dx} = 3x^2$	u(0)=1
3	$\frac{dy}{dx} = 7x + 6$	u(0)=3

Table 1: List of problem solved by PINN

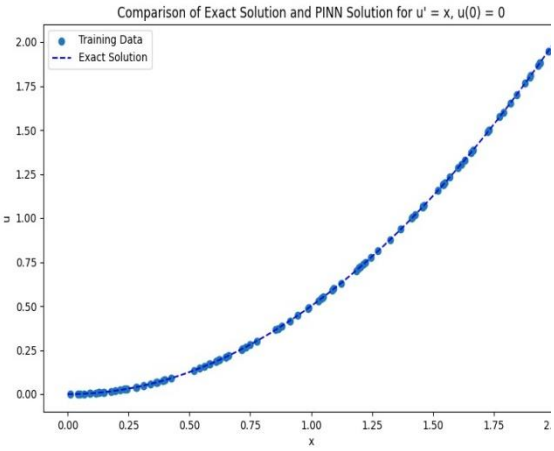

Figure: 2

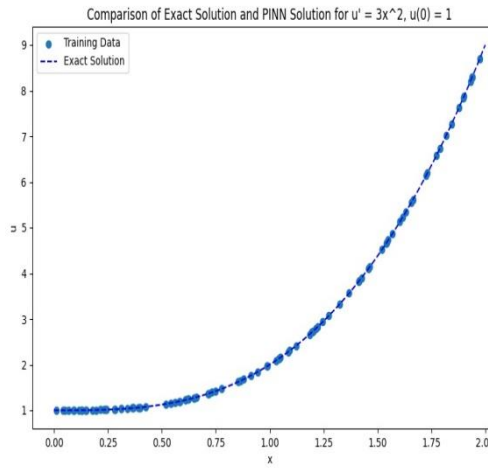

Figure:3

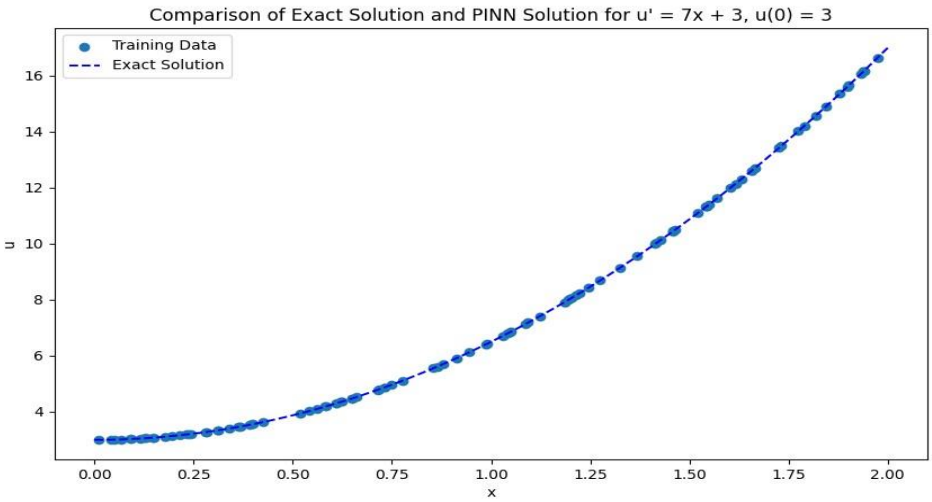

Figure:4

II Non-Linear Differential Equation

Our attention is directed towards the logistic differential equation, a prominent first-order ordinary differential equation utilized for the modelling of population growth. The familiar expression for logistic growth is given by:

$$\frac{df}{dt} = R\, f(t)(1 - f(t)) \qquad \ldots (1)$$

In this context, the function f(t) signifies the population growth rate as a function of time t, and the parameter R determines the maximum population growth rate, significantly influencing the solution's shape.

Given the initial condition f (0) =f_0 , the solution to equation (1) can be expressed as:

$$f(t) = \frac{f_0}{f_0 + (1 - f_0)e^{-t}}$$

If $f_0 = \frac{1}{2}$, then the solution is the logistic function

$$f(t) = \frac{1}{1 + e^{-t}}$$

a. Solution with PINN

The solution of an Ordinary Differential Equation (ODE) through the Physics-Informed Neural Network (PINN) entails harnessing neural network capabilities while integrating physical constraints to ensure accurate predictions. The process involves:

- The neural network
- Constructing the loss function
- Solving the differential equation with PINNs

Analytically deriving the solution to the logistic differential equation (1) serves as a straightforward illustration of how PINNs work. All the techniques elucidated in the subsequent sections are easily adaptable to more intricate ordinary differential equations. PINNs rely on two fundamental properties of neural networks (NNs):

1. Universal Function Approximators:

Demonstrations from formal studies [12,13] affirm that neural networks (NNs) function as universal approximators for any given function. Consequently, a sufficiently deep and expressive NN can effectively approximate any function, including solutions to the provided differential equation.

2. Automatic Differentiation:

The computation of derivatives (of any order) for an NN output concerning its inputs (and, naturally, model parameters during backpropagation) is straightforward through automatic differentiation (AD). In fact, it is AD that has been instrumental in making neural networks efficient and successful from the outset.

The most crucial contribution to the loss is determined as the residual of the differential equation, expressed as follows:

$$\frac{df_{NN}}{dt} - Rf_{NN}(t)\big(1 - f_{NN}(t)\big) = 0$$

where $f_{NN}(t)$ is the output of a NN with one input and its derivative is computed using AD. It is readily apparent that if the neural network output respects the aforementioned equation, one

is effectively solving the differential equation. To quantify the specific loss contribution arising from the residual of the differential equation, it is necessary to define a set of points within the equation domain, commonly known as collocation points. The mean square error (MSE) or an alternative loss function is then evaluated as an average over all the selected collocation points:

$$L_{DE} = \frac{1}{M}\sum_{j=1}^{M}\left(\frac{df_{NN}}{dt}|_{t_j} - Rf_{NN}(t_j)(1 - f_{NN}(t_j))\right)^2$$

The loss contribution is determined by averaging the residual of the differential equation over a set of collocation points. To incorporate the boundary condition, it is included in the loss computation in the same way as described previously:

$$L_{BC} = (f_{NN}(t_0) - 0.5)^2 \text{ with } t_0 = 0$$

This adds the loss contribution from the boundary conditions to the Mean Squared Error (MSE) loss. Consequently, the final loss is a sum of the losses from the differential equation and the boundary conditions:

$$L = L_{DE} + L_{BC}$$

Throughout the optimization process, this combined loss is minimized, leading to the training of the neural network output to adhere to both the differential equation and the provided boundary condition, thereby approximating the final solution to the differential equation.

a. Analytical Solution (Laplace Transform Method)

A function $f: (0, \infty) \rightarrow F$ is considered to be of exponential order if there exists a constant M (>0) $\in R$ such that $|f(t)| \leq Me^{at}$ for all $t>0$. For each function $f:(0, \infty) \rightarrow R$ of exponential order, we define the Laplace transform of the function as:

$$L(f(t)) = F(s) = \int_0^{\infty} f(t)e^{-st}dt.$$

The Laplace transform of the function f is represented as L(f). It is also established that L is a linear and injective operator. Consequently, at points where f is continuous, we have:

$$f(t) = L^{-1}(F(s)) = \int_0^{\infty} f(t)e^{st}dt$$

this is called the inverse Laplace transforms.

If L(f(t)) is Laplace transform of f(t) , then

i) $L(1) = \frac{1}{s}$

ii) $L(t^n) = \frac{n!}{s^{n+1}}$

iii) $L(e^{at}) = \frac{1}{s-a}$

iv) $L(f'(t)) = sF(s) - f_0$.

We consider the logistic differential equation

$$\frac{df}{dt} = f(t) - g(f), \qquad t \geq 0 \qquad \text{….. (2)}$$

with initial condition $f(t = 0)$=0.5. Let where g is a nonlinear function of f.

By applying Laplace transform to both sides of differential equation (2), we obtain:

$$sF(s) - f_0 = F(s) - G(s)$$

$$F(s) = \frac{f_0}{s-1} - \frac{G(s)}{s-1} \qquad \text{……(3)}$$

Therefore, assuming the inverse Laplace transform to (3) we obtain

$$f(t) = f_0 \exp(t) - L^{-1}\left(\frac{G(s)}{s-1}\right) \qquad \text{….. (4)}$$

Let $g(f) = f^2$, then by power series

$$f(t) = \sum_{n=1}^{\infty} a_n t^n, \qquad \text{…… (5)}$$

we obtain

$$g(f) = \left(\sum_{n=1}^{\infty} a_n t^n\right)^2 = a_0^2 - 2a_0a_1t + (2a_0a_2 + a_1^2)t^2 + \ldots.$$

By Laplace transform

$$G(s) = \frac{a_0^2}{s} + \frac{2a_0a_1}{s^2} + \frac{4a_0a_2+2a_1^2}{s^3} + \frac{12a_0a_3+12a_1a_2}{s^4} + \ldots.$$

Using (4) one get

$$F(s) = \frac{0.5}{t-1} - \left\{\frac{a_0^2}{s(s-1)} + \frac{2a_0a_1}{s^2(s-1)} + \frac{4a_0a_2+2a_1^2}{s^3(s-1)} + \frac{12a_0a_3+12a_1a_2}{s^4(s-1)} + \ldots..\right\}$$

Applying the inverse Laplace transform to this equation yield:

$$a_0 + a_1t + a_2t^2 + a_3t^3 + \ldots. = 0.5\left(1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots\right) - a_0^2t - \left(\frac{a_0^2}{2} - a_0a_1\right)t^2 - \left(\frac{a_0^2}{6} + \frac{a_0a_1}{3} + \frac{2a_0a_2}{3} + \frac{a_1^2}{3}\right)t^3 - \ldots.$$

$$= 0.5 + (0.5 - a_0^2)t + \left(\frac{1}{4} - \frac{a_0^2}{2} - a_0a_1\right)t^2 + \left(\frac{1}{12} - \frac{a_0^2}{6} - \frac{a_0a_1}{3} - \frac{2a_0a_2}{3} - \frac{a_1^2}{3}\right)t^3 + \ldots.$$

Equating the coefficient of power t gives

$$a_0 = 0.5, \; a_1 = 0.5 - a_0^2 \Longrightarrow a_1 = 0.25,$$

$$a_2 = \frac{1}{4} - \frac{a_0^2}{2} - a_0a_1 \Longrightarrow a_2 = 0,$$

$$a_3 = \frac{1}{12} - \frac{a_0^2}{6} - \frac{a_0a_1}{3} - \frac{2a_0a_2}{3} - \frac{a_1^2}{3} \Longrightarrow a_3 = \frac{1}{48}, \ldots.$$

the solution u(t) derived from equation (5) is expressed as follows:

$$f(t) = 0.5 + 0.25t + 0.2t^3 + \cdots,$$

representing the obtained solution.

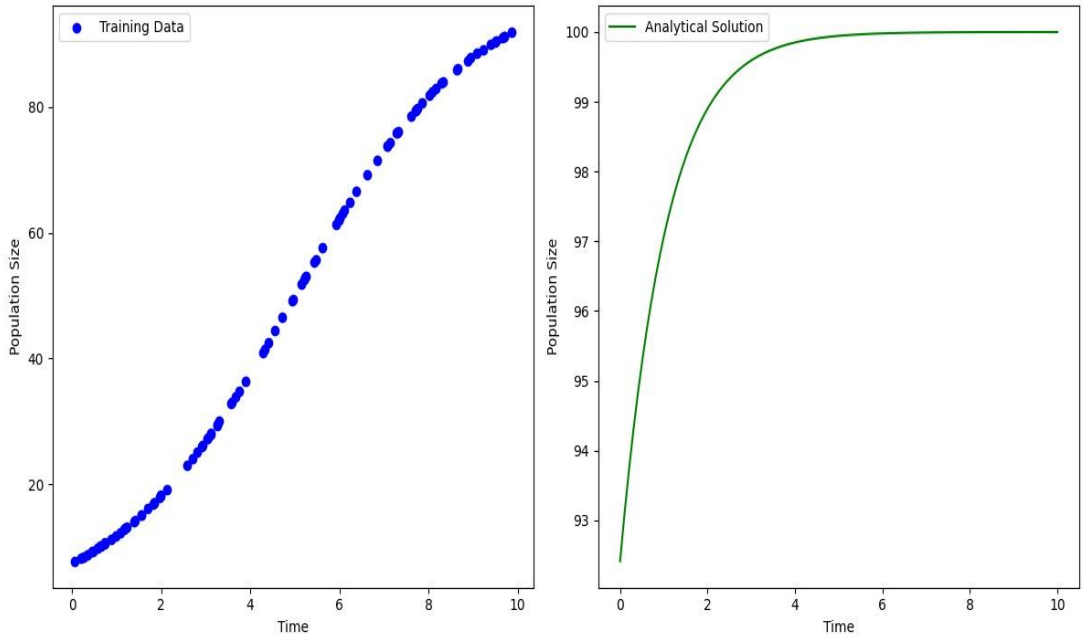

Figure :5 PINN solution acquired for the logistic differential equation with epochs=1000 (Left) and analytical solution obtained using Laplace transform (Right).

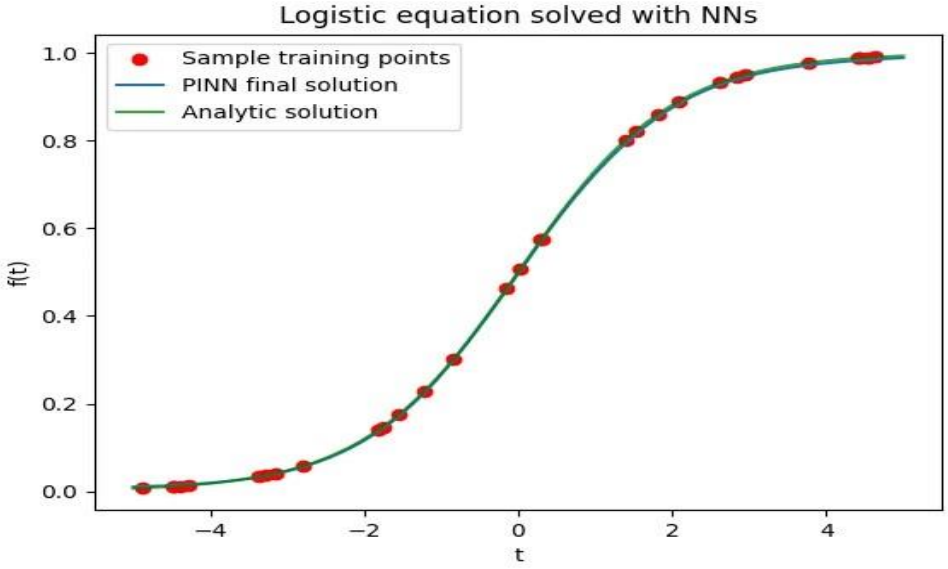

Figure:6 The application of the PINN approach to solve the logistic differential equation, accompanied by the depiction of a set of randomly chosen training points.

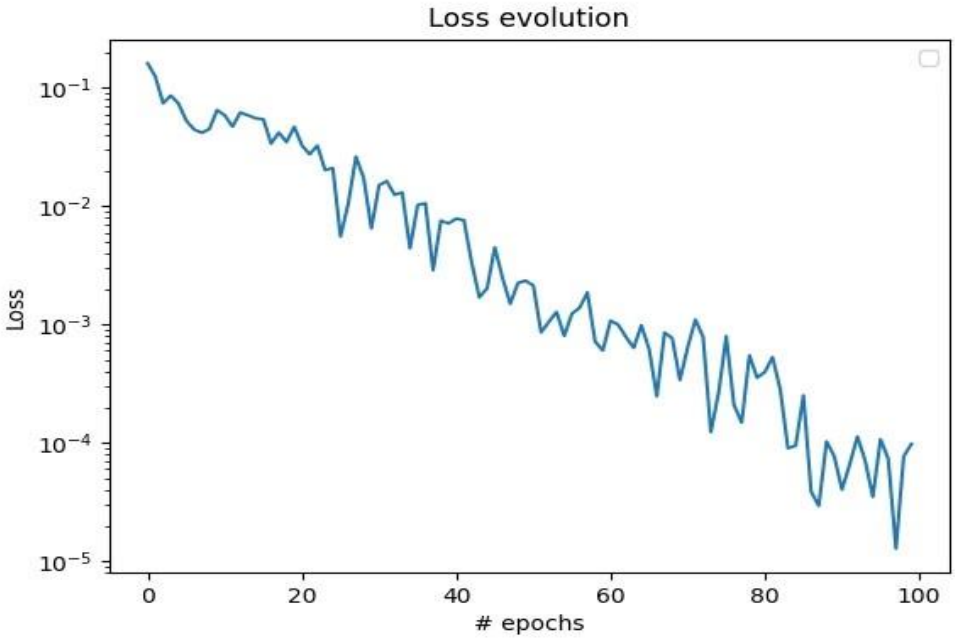

Figure:7 The estimate of loss function values using Adam optimizer.

4. Conclusions

This study introduces a Physics-Informed Neural Network (PINN) as an alternative for solving differential equations. Through a comparison with exact and analytic methods, the research highlights the effectiveness of PINNs in capturing complex dynamics across diverse applications. The study's carefully crafted graphs underscore the accuracy and efficiency of the PINN methodology. By eliminating the need for explicit analytical solutions, PINNs provide a flexible framework with the potential to revolutionize problem-solving in scientific and engineering domains. This research contributes valuable insights into the capabilities of PINNs, emphasizing their role in advancing computational methodologies for differential equations.

References

1. Abirami, P. Prakash and K. Thangavel, Fractional diffusion equation based image denoising using CN-GL scheme, International Journal of Computer Mathematics, vol. 95, pp. 1222–1239, 2018.
2. A. Abirami, P. Prakash and Y.K. Ma Variable-Order Fractional Diffusion ModelBased Medical Image Denoising, Mathematical Problems in Engineering, Hindawi, Volume(2021), Article ID 8050017, 10 pages
3. Abdoonand, M. A., and Hasan, F. L., (2022), Advantages of the Differential Equations for Solving Problems in Mathematical Physics with Symbolic Computation, Mathematical

Modelling of Engineering Problems, 9(1), 268.
4. Overstall, A. M., Woods D. C., and Parker, B. M., (2019), Bayesian Optimal Design for Ordinary Differential Equation Models with Application in Biological Science, Journal of the American Statistical Association, pp. 583 - 598.
5. Gallegos, R. R., (2015), A Differential Equations Course for Engineers through Modelling and Technology, Mathematical Modelling in Education Research and Practice, pp. 545 - 555.
6. Luo, J., Abdullah, F., and P. D. Christofides, (2023), Model Predictive Control of Nonlinear Processes Using Neural Ordinary Differential Equation Models, Computers & Chemical Engineering, 178, 108367.
7. Sirignano, J., and Spiliopoulos, K., (2018), Dgm: A Deep Learning Algorithm for Solving Partial Differential Equations, Journal of Computational Physics, 375, pp. 1339 - 1364.
8. Michoski, C., Milosavljevic, M., Oliver, T., and Hatch, D. R., (2020), Solving differential equation using deep learning networks, Neurocomputing, 399, pp. 193 - 212.
9. Katsikis, D., Muradova, A. D., and Stavroulakis, G. E., (2022), A Gentle Introduction to Physics - Informed Neural Network, with Applications in Static and Beam Problems, 9 pp. 103 - 128.
10. LeCun, Y., Bengio, Y., and Hinton, G., (2015), Deep Learning, Nature, 521, pp. 436 - 444.
11. Raissi, M., Perdikaraisand, P., EmKarniadakis, G., (2017), Physics Informed Deep Learning (Part 1): Data-driven Solutions of Nonlinear Partial Differential Equations, arXiv>cs>arXiv: 1711.10561[Cs:AI].
12. Baydlin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M., (2018), Automatic Diff erentiation in Machine Learning, arXiv:1502.05767v4[cs.SC].
13. Hornik, K., Stinchcombe, M., and White, H., (1989), Multilayer feed forward networks are universal approximations, Neural Networks 2, 359366.
14. Karniadakis, G. E., Ioannis G. Kevrekidis, Paris Perdikaris, L. L., Sifan Wang and Liu Yang, (2021), Physics informed machine learning, Nature Reviews Physics 3, 422440