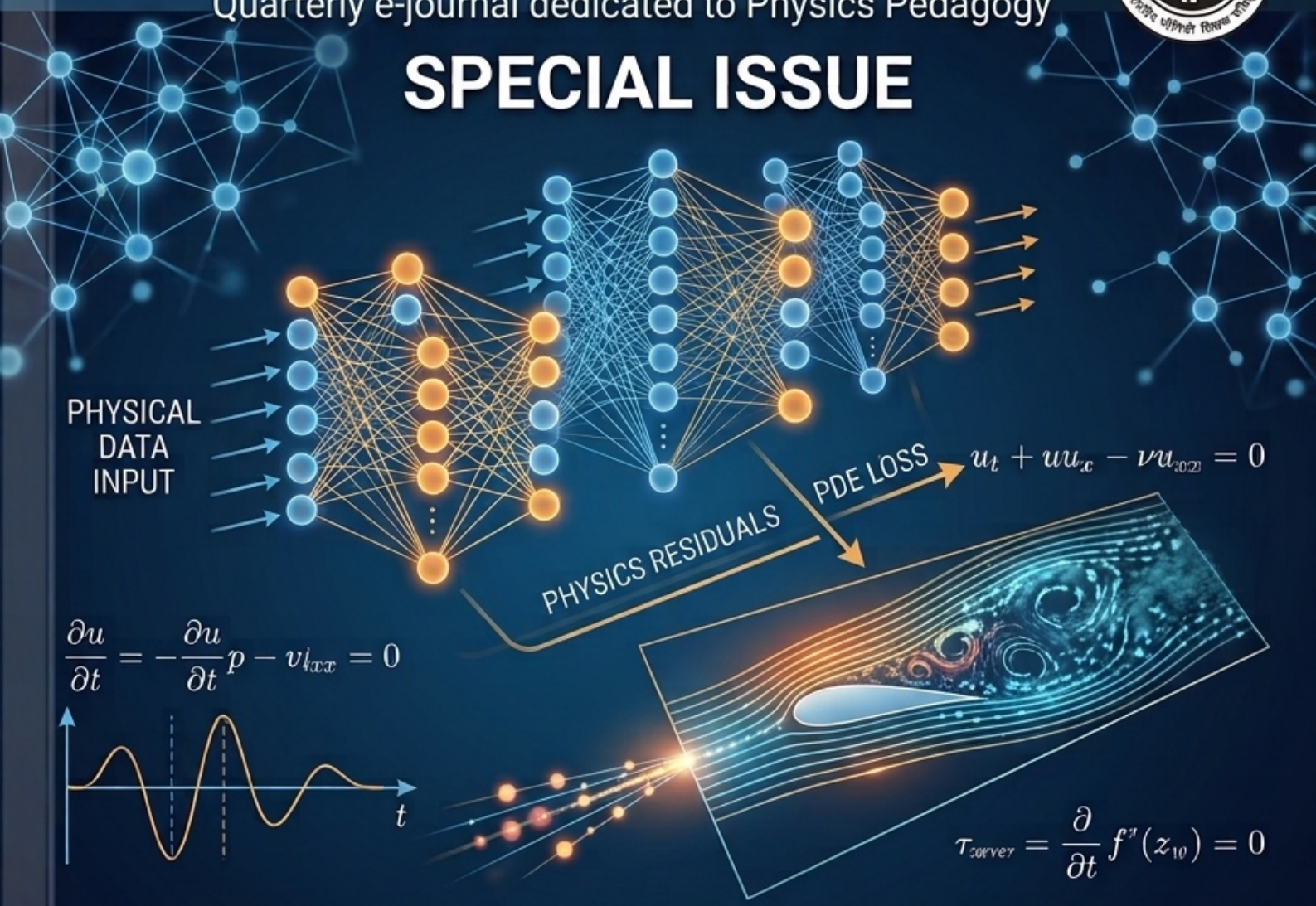
Physics Education
Quarterly e-journal dedicated to Physics Pedagogy

SPECIAL ISSUE



PHYSICS-INFORMED NEURAL NETWORKS (PINNs)

International Conference-cum-Round Table on Translational Research and Innovation in Beam Technologies, February 13-15 2026

Volume 40 - Special Issue

Integrating Physical Principles with Artificial Intelligence: Theory, Applications, and Education

JOURNAL OF PHYSICS EDUCATION
VOLUME 40 • ICTRIBT 2026 / SPECIAL ISSUE / 1

EDITORIAL

Bridging Physical Laws and Machine Intelligence: Physics-Informed Neural Networks in Research and Pedagogy

A Preface to the Special Issue on Physics-Informed Neural Networks (PINNs)

In recent years, the convergence of computational physics and artificial intelligence has given rise to a transformative paradigm: Physics-Informed Neural Networks (PINNs). By embedding fundamental physical principles—such as conservation laws, non-linear partial differential equations (PDEs), and boundary conditions—directly into the loss function of deep neural architectures, PINNs bridge the traditional divide between purely data-driven machine learning and physics-based analytical modeling. This Special Issue of the *Journal of Physics Education* (Vol. 40, ICTRIBT 2026/Special Issue/1) brings together a compelling collection of eight research articles showcasing the pedagogical, conceptual, and practical application of PINNs across diverse physics domains.

> **Core Theme:** Incorporating mathematical conservation laws into deep learning frameworks not only regularizes neural network training but also provides highly intuitive computational tools for physics education and research.

HIGHLIGHTS OF THE SPECIAL ISSUE

The contributed manuscripts in this volume span key disciplines in theoretical, computational, and applied physics, illustrating the versatility of the PINN paradigm:

1. **Quantum Mechanics & Wave Equations:** Quantum systems provide a fertile ground for PINN applications. *Thakur and Awasthi* demonstrate the solution of the time-independent Schrödinger equation for double-well potentials in quantum plasma systems (pp. 1–8). Complementing this, *Walia and Awasthi* explore the time-dependent Schrödinger equation for harmonic oscillators while extracting the Bohm quantum potential (pp. 1–12). Furthermore, *Saklani and Awasthi* apply PINNs to solve the Woods-Saxon potential to elucidate the shell structure of ^{40}Ca (pp. 1–8), and *Kanika et al.* determine the ground state of the hydrogen atom (pp. 1–8).

2. **Molecular Dynamics & Nuclear Physics:** Expanding beyond fundamental wave equations, *Kashyap et al.* employ PINNs to determine the vibrational energy levels of the HCl molecule (pp. 1–7), highlighting molecular spectroscopy applications. In nuclear physics, *Anjali and Awasthi* utilize PINNs to estimate the coefficients of the Semi-Empirical Mass Formula (SEMF) (pp. 1–10), offering a fresh computational perspective on nuclear binding energy.

3. **Thermal & Differential Systems:** Addressing fundamental transport phenomena, *Rathore et al.* present a PINN-based numerical formulation for the classic heat equation (pp. 1–10). In a broader mathematical context, *Sharma et al.* establish general frameworks for learning solutions of coupled differential equations using physics-informed architectures (pp. 1–6).

EDUCATIONAL SIGNIFICANCE & OUTLOOK

Beyond their numerical efficacy, PINNs hold tremendous potential for physics pedagogy. They empower students and researchers to explore non-linear, multi-dimensional, and inverse problems that were previously intractable using standard introductory software. By demystifying deep learning through the lens of physical conservation laws, this special issue serves both as a milestone in computational research and a practical roadmap for modernizing physics education curricula.

Editorial Team, Journal of Physics Education

JOURNAL OF PHYSICS EDUCATION
VOLUME 40 • ICTRIBT 2026 / SPECIAL ISSUE / 1

SPECIAL ISSUE ON
Physics-Informed Neural Networks (PINNs)

TABLE OF CONTENTS / INDEX OF MANUSCRIPTS

1. **Solving Time-Independent Schrödinger Equation for a Double-Well Potential in Quantum Plasma Systems using Physics-Informed Neural Networks** pp. 1–8
Tanisha Thakur and Ayushi Awasthi

2. **Solving the Time-Dependent Schrödinger Equation for a Harmonic Oscillator and Extracting the Bohm Quantum Potential Using PINNs** pp. 1–12
Sanyam Walia and Ayushi Awasthi

3. **Physics Informed Neural Networks (PINNs) Approach for Numerical Solution of Heat Equation** pp. 1–10
Gargi Rathore, Arushi Sharma and Ishwar Kant

4. **Learning Solutions of Coupled Differential Equations with Physics-Informed Neural Networks** pp. 1–6
Aman Sharma, Ayushi Awasthi and O.S.K.S. Sastri

5. **Physics-Informed Neural Networks (PINNs) for the Determination of Vibrational Energy Levels of the HCl Molecule** pp. 1–7
Tanuja Kashyap, Arushi Sharma and Ishwar Kant

6. **Physics Informed Neural Network Approach for Estimating the Coefficients of the Semi-Empirical Mass Formula** pp. 1–10
Anjali and Ayushi Awasthi

7. **Shell Structure of ^{40}Ca from the Woods-Saxon Potential by Solving the Time Independent Schrödinger Equation Using PINNs** pp. 1–8
Tamanna Saklani and Ayushi Awasthi

8. **Solving Hydrogen Atom using Physics-Informed Neural Network for Ground State** pp. 1–8
Kanika, Ishwar Kant, Ayushi Awasthi and Arushi Sharma

Journal's mission: *Integrating Physical Principles with Artificial Intelligence: Theory, Applications, and Education.*

Solving Time-Independent Schrodinger Equation for a Double-Well Potential in Quantum Plasma Systems using Physics-Informed Neural Networks

Tanisha Thakur and Ayushi Awasthi

Department of Physics and Astronomical Sciences
Central University of Himachal Pradesh, Dharamshala (176215), H.P. (Bharat).
tanishat287@gmail.com

Submitted on 05-Feb-2026

Abstract

The time-independent Schrödinger equation plays a fundamental role in describing quantum tunneling phenomena in systems governed by double-well potentials, which arise naturally as effective potentials in quantum plasma environments due to external fields and confinement effects. In this work, we employ Physics-Informed Neural Networks (PINNs) to solve one-dimensional Schrödinger equation for a symmetric double-well potential and obtain accurate ground and low-lying excited state solutions. The fully connected neural networks is used to obtain the wavefunctions, while the energy eigenvalues are treated as trainable parameters and determined through the minimization of a composite loss function enforcing the governing differential equation, boundary conditions, normalization, and orthogonality constraints. Even and odd-parity network architectures are used to directly capture symmetric and antisymmetric eigenstates. The PINNs results are validated against finite-difference matrix diagonalization technique, showing good agreement for the lowest energy levels and correctly reproducing the near-degeneracy caused by quantum tunneling between the wells. These results demonstrate that PINNs provide a flexible, accurate, and mesh-free alternative to conventional numerical techniques, making them well suited for modeling tunneling phenomena and effective quantum potentials in plasma-related systems.

1 Introduction

The time-independent Schrödinger equation plays a central role in quantum mechanics, governing the bound-state structure and tunneling dynamics of microscopic systems [1]. Among the class of exactly nontrivial quantum potentials, the symmetric double-well potential [2] occupies a distinguished

position due to its relevance in molecular inversion phenomena [3], quantum tunneling [4, 5], energy-level splitting [6] and plasma-modified quantum confinement [7]. In plasma environments, screening effects and collective interactions substantially modify the effective potential landscape, leading to rich quantum behavior that demands accurate and flexible computational approaches [8].

Traditionally, the Schrödinger equation for double-well systems has been solved using numerical techniques such as finite difference schemes, matrix mechanics [2], spectral methods [9], and shooting algorithms [10]. While these methods are well established and reliable, they often require dense spatial grids, careful treatment of boundary conditions, and repeated eigenvalue solvers to extract multiple states. Moreover, extending these approaches to plasma-modified or parameter-dependent potentials significantly increases computational cost and complexity [7].

In recent years, Physics-Informed Neural Networks (PINNs) have emerged as a powerful mesh-free framework for solving differential equations by embedding the governing physical laws directly into the training process of neural networks [11]. Unlike purely data-driven models, PINNs incorporate the Schrödinger equation as a soft constraint through automatic differentiation, enabling the simultaneous learning of wavefunctions and energy eigenvalues without discretizing the spatial domain [12]. This paradigm offers key advantages, including continuous representations of solutions, natural enforcement of physical constraints, and scalability to complex or parameterized potentials.

In this work, we employ the PINNs framework to solve the one-dimensional time-independent Schrödinger equation for a symmetric double-well potential relevant to quantum plasma systems. By explicitly incorporating parity symmetry, normalization, and orthogonality conditions into the loss function, the proposed approach accurately captures tunneling-induced near-degeneracy and reproduces low-lying energy eigenstates. The results demonstrate that PINNs provide an efficient and flexible alternative to traditional numerical solvers, particularly suited for plasma-modified quantum systems where adaptability and physical consistency are essential.

2 Methodology

The present study is based on the time-independent Schrödinger equation describing a quantum particle moving in an effective one-dimensional potential [13],

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x), \quad (1)$$

where $\psi(x)$ is the stationary wavefunction, E denotes the energy eigenvalue, and $V(x)$ represents the external potential [1]. In order to reduce the number of free physical parameters and improve numerical stability,

the equation is expressed in dimensionless form by adopting atomic units $\hbar = 1$ and $m = 1$. This leads to

$$-\frac{1}{2}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x). \quad (2)$$

Choice of Potential

To model tunneling and level splitting phenomena relevant to quantum plasma systems, we consider a symmetric double-well potential of the form [14]

$$V(x) = (ax^2 - b^2)^2 \quad (3)$$

where the parameters a and b control the curvature of the wells and the barrier height, respectively. This quartic form naturally produces two equivalent minima separated by a finite barrier and is widely used to study tunneling-induced near-degeneracy in bound states [15]. In the present work, the parameters are chosen as $a = 1$ and $b = 1.5$, ensuring well-separated minima and a finite tunneling probability for low-lying states.

Physics-Informed Neural Networks (PINNs) Framework

Within the Physics-Informed Neural Network (PINNs) framework [16], the wavefunction $\psi(x)$ is approximated by a fully connected neural network $\mathcal{N}(x;\theta)$ [17], where θ denotes the trainable parameters. Unlike conventional numerical solvers, the energy eigenvalues are treated as independent trainable parameters and learned simultaneously with the wavefunctions.

To explicitly enforce the parity symmetry of the double-well potential, separate neural architectures are employed for even and odd eigenstates [18], defined as

$$\begin{aligned} \psi_{\text{even}}(x) &= \mathcal{N}(x) + \mathcal{N}(-x), \\ \psi_{\text{odd}}(x) &= \mathcal{N}(x) - \mathcal{N}(-x). \end{aligned} \quad (4)$$

This construction ensures that the learned eigenfunctions possess the correct symmetry properties without additional constraints.

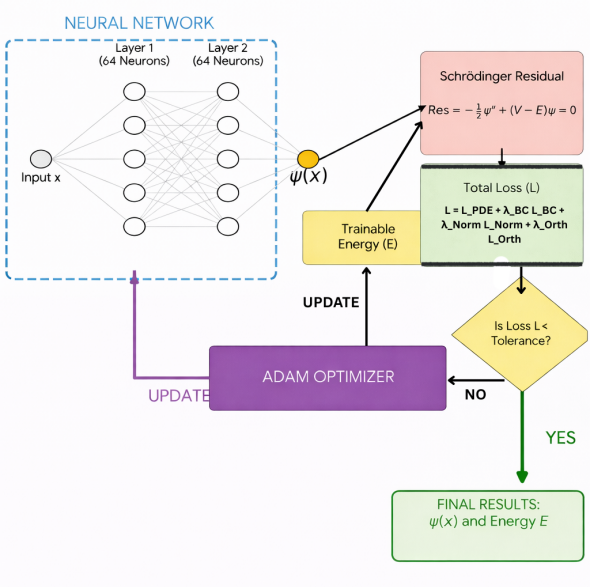


Figure 1: Workflow of the Physics-Informed Neural Network (PINNs) for solving the one-dimensional Schrödinger equation.The neural network approximates the wavefunction $\psi(x)$ while the energy E is treated as a trainable parameter, optimized using the ADAM optimizer by minimizing the total loss.

Loss Function and Training

The physics-informed neural network (PINNs) [19] is trained by minimizing a composite loss function that enforces the governing Schrödinger equation along with the physical constraints of the quantum system. The total loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}} + \lambda_{\text{Norm}}\,\mathcal{L}_{\text{Norm}} + \lambda_{\text{Orth}}\,\mathcal{L}_{\text{Orth}}, \tag{5}$$

where the physics loss term enforces the time-independent Schrödinger equation and is given by

$$r_n(x) = -\frac{1}{2}\frac{d^2\psi_n(x)}{dx^2} + V(x)\psi_n(x) - E_n\psi_n(x) \tag{6}$$

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_c}\sum_{n=0}^{N_s-1}\sum_{i=1}^{N_c} |r_n(x_i)|^2 \tag{7}$$

The boundary condition loss ensures the wavefunctions vanish at the domain boundaries and is expressed as

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_s}\sum_{n=0}^{N_s-1}\left(|\psi_n(-L)|^2 + |\psi_n(L)|^2\right). \tag{8}$$

To guarantee physical admissibility, the normalization loss enforces unit norm of each eigenstate,

$$\mathcal{L}_{\text{Norm}} = \frac{1}{N_s}\sum_{n=0}^{N_s-1}\left|\int_{-L}^{L}|\psi_n(x)|^2\,dx - 1\right|^2, \tag{9}$$

while the orthogonality loss ensures mutual orthogonality between distinct eigenfunctions,

$$\mathcal{L}_{\text{Orth}} = \frac{1}{N_s(N_s-1)}\sum_{\substack{m,n=0\\ m\neq n}}^{N_s-1}\left|\int_{-L}^{L}\psi_m(x)\psi_n(x)\,dx\right|^2. \tag{10}$$

The weighting parameters λ_{BC}, λ_{Norm}, and λ_{Orth} control the relative importance of each constraint during training. Equal weights $\lambda_{\text{BC}} = \lambda_{\text{Norm}} = \lambda_{\text{Orth}} = 10$ are used throughout the training.

The neural network consists of two hidden layers with 64 neurons each and hyperbolic tangent (*tanh*) activation functions. Training is performed using the Adam optimizer [20] with a learning rate of 10^{-3} over 10^4 epochs. Uniform collocation points are employed over the spatial domain, and automatic differentiation [21] is used to compute the second-order spatial derivatives.

3 Results and Discussion

The Physics-Informed Neural Network (PINNs) framework was applied to solve the time-independent Schrödinger equation for a symmetric double-well potential [22]. The model employed two hidden layers with 64 neurons each, using smooth activation functions to ensure differentiability of the wavefunction. Equal weights were assigned to the normalization, orthogonality, and boundary condition loss terms ($\lambda_{\text{Norm}} = \lambda_{\text{Orth}} = \lambda_{\text{BC}} = 10$), and the network was trained for 10^4 epochs for the three lowest eigenstates ($n = 0, 1, 2$).

The convergence behaviour of the energy

eigenvalues during training indicates stable and systematic learning. As training progresses, the total loss reduces by nearly three orders of magnitude, reflecting the simultaneous enforcement of the Schrödinger equation, boundary conditions, normalization, and orthogonality constraints. The final PINN-predicted energies are summarized below:

Table 1: Comparison of FD and PINN energy eigenvalues.

State	$E_{\mathrm{FD}}(hartree)$	$E_{\mathrm{PINN}}(hartree)$
$n = 0$	1.9705	1.9709
$n = 1$	2.0123	2.1500
$n = 2$	4.9079	4.9087

The ground state ($n = 0$) and second excited state ($n = 2$) energies obtained via PINNs are in excellent agreement with the finite difference (FD) results, with relative deviations below the percent level. The first excited state ($n = 1$) shows a comparatively larger deviation, which can be attributed to the increased sensitivity of higher-energy states to orthogonality enforcement and loss balancing during training. The Schrödinger equation is solved in dimensionless natural units by setting $\hbar = 1$ and $m = 1$, so the computed eigenvalues are energies in natural units that is the hartree (Ha).

The predicted wavefunctions using PINNs is compared with those obtained using the finite difference (FD) method as shown in figure 2. The excellent agreement in both amplitude and spatial structure demonstrates that the PINNs accurately captures

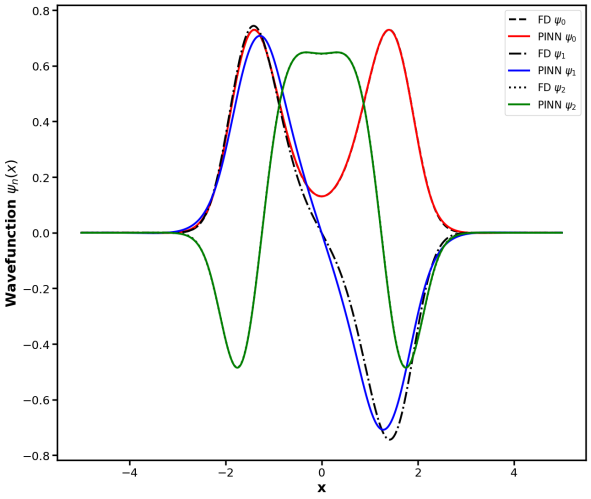


Figure 2: Comparison of eigenfunctions obtained using the Physics-Informed Neural Network (PINNs) and the Finite Difference (FD) method for the first three quantum states of the symmetric double-well potential. Solid colored curves represent PINNs solutions, while black dashed, dash-dotted, and dotted curves correspond to FD solutions for the ground, first excited, and second excited states respectively.

the nodal structure and tunnelling behaviour across the potential barrier. In particular, the smooth penetration of the wavefunctions into the classically forbidden region clearly illustrates quantum tunnelling, a hallmark feature of the double-well system.

The PINNs predicted low-lying eigenstates superimposed on the symmetric double-well potential is illustrated in figure 3. The wavefunctions are vertically shifted by their corresponding energy eigenvalues for clarity. The ground state ψ_0 exhibits even par-

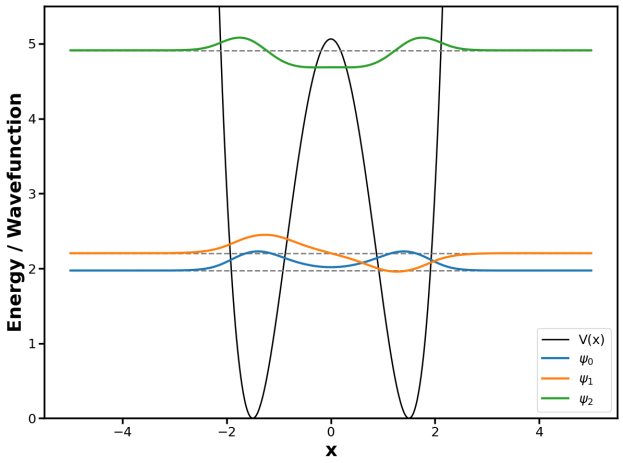


Figure 3: PINNs-computed eigenstates superposed on the symmetric double-well potential. The nonzero amplitude of the wavefunctions within the barrier region confirms quantum tunneling and results in a small energy splitting between the lowest two states.

ity with a finite amplitude at the center of the barrier, indicating strong tunnelling between the two wells [23]. The first excited state ψ_1 is antisymmetric, with a node at the center, as required by parity symmetry, while still displaying appreciable penetration into the classically forbidden region. The second excited state ψ_2 shows reduced tunnelling and increased localization within individual wells due to its higher energy.

The small energy separation between ψ_0 and ψ_1 reflects the lifting of degeneracy caused by quantum tunnelling through the finite potential barrier. This splitting would vanish in the limit of an infinitely high barrier, corresponding to fully localized states. The smooth and physically consistent behaviour of all eigenstates demonstrates the ability of the PINN framework to capture subtle quantum effects such as parity symmetry, tunnelling, and near-degeneracy without explicit basis construction or grid discretization.

An important physical feature of the symmetric double-well potential is the near-degeneracy of the ground and first excited states [22]. In the limit of an infinitely high barrier, these states would be exactly degenerate, corresponding to states localized in the left and right wells. However, due to finite barrier height, tunnelling lifts this degeneracy, resulting in a small energy splitting between the symmetric (even-parity) ground state and the antisymmetric (odd-parity) first excited state [24]. The PINN framework successfully captures this splitting, confirming its ability to resolve subtle quantum effects arising from wavefunction overlap across the barrier. This agreement also validates the parity-constrained network design used for the low-lying eigenstates.

Overall, these results demonstrate that PINNs provide a robust and flexible alternative to traditional numerical techniques [2, 10] such as finite difference and matrix mechanics methods. Unlike grid-based approaches, PINNs yield smooth, analytical representations of eigenfunctions, naturally incorporate physical constraints, and scale efficiently to higher-dimensional and more complex quantum systems.

4 Conclusion:

PINNs accurately solved the double-well Schrödinger problem using physics-based loss constraints and no spatial grid. The method reproduced eigenstates, tunneling behavior, and energy splitting in good agreement with finite-difference results, showing that PINNs are an efficient and flexible tool for quantum potential problems. However the present formulation assumes a one-dimensional, time-independent system and neglects explicit many-body and relativistic effects.

5 Acknowledgments

A. Awasthi acknowledges the financial support provided by the DST, Govt. of India, through Grant No. DST/INSPIRE Fellowship/2020/IF200538.

References

[1] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. Cambridge University Press, 3 edition, 2018.

[2] Vedran Jelić and Frank Marsiglio. The double-well potential in quantum mechanics: a simple, numerically exact formulation. *European Journal of Physics*, 33(6):1651–1666, 2012.

[3] Ha Vinh Lam Nguyen, Iwona Gulaczyk, Marek Kreglewski, and Isabelle Kleiner. Large amplitude inversion tunneling motion in ammonia, methylamine, hydrazine, and secondary amines: From structure determination to coordination chemistry. *Coordination Chemistry Reviews*, 436:213797, 2021.

[4] Johann Förster, Etienne Plesiat, Alvaro Magana, and Alejandro Saenz. Imaging of the umbrella motion and tunneling in ammonia molecules by strong-field ionization. *Physical Review A*, 94:043405, 2016.

[5] Caio M. Porto and Nelson H. Morgon. Analytical approach for the tunneling process in double well potentials using irc calculations. *Computational and Theoretical Chemistry*, 1187:112917, 2020.

[6] Csaba Fábri, Roberto Marquardt, Attila G. Császár, and Martin Quack. Controlling tunneling in ammonia isotopomers. *The Journal of Chemical Physics*, 150(1):014102, 2019.

[7] Madhulita Das, B. K. Sahoo, and S. Pal. Plasma-screening effects on the electronic structure of multiply charged al ions using debye and ion-sphere models. *Physical Review A*, 93:052513, 2016.

[8] J. C. Stewart and K. D. Pyatt. Lowering of ionization potentials in plasmas. *The Astrophysical Journal*, 144:1203–1211, 1966.

[9] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover, 2 edition, 2001.

[10] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.

[11] Haojie Jin, Marios Mattheakis, and Pavlos Protopapas. Solving quantum eigenvalue problems using physics-informed neural networks. *Neural Networks*, 134:105–117, 2021.

[12] Soumyadip Sarkar. Physics-informed neural networks for one-dimensional quantum well problems. *arXiv preprint arXiv:2504.05367*, 2025.

[13] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2 edition, 2017.

[14] Sidney Coleman. *Aspects of Symmetry: Selected Erice Lectures*. Cambridge University Press, 1985.

[15] A. M. Dyugaev and P. D. Grigoriev. Modeling of double-well potentials for the schrödinger equation. *Journal of Experimental and Theoretical Physics*, 137(1):17–22, 2023.

[16] F. Marsiglio. The harmonic oscillator in quantum mechanics: A third way. *American Journal of Physics*, 77(3):253–258, 2009.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[18] R. Shankar. *Principles of Quantum Mechanics*. Springer, 2 edition, 1994.

[19] Maziar Raissi, Paris Perdikaris, and George 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*, 378:686–707, 2019.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[21] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

[22] Yi-Qiang Wu, Xuan Liu, Hanlin Li, and Fuqiang Wang. Energy-embedded neural solvers for one-dimensional quantum systems. *arXiv preprint arXiv:2505.24194*, 2025.

[23] Danilo F. Schafaschek, Giovani L. Vasconcelos, and Antônio M. S. Macêdo. Tunneling in double-well potentials within stochastic quantization: Application to ammonia inversion. *arXiv preprint arXiv:2512.16168*, 2025.

[24] Eugen Merzbacher. *Quantum Mechanics*. Wiley, 3 edition, 1998.

Solving the Time-Dependent Schrödinger Equation for a Harmonic Oscillator and Extracting the Bohm Quantum Potential Using PINNs

Sanyam Walia[1] and Ayushi Awasthi[1]

[1]Department of Physics and Astronomical Science,
Central University of Himachal Pradesh, Dharamshala, India
sanyamwalia1310@gmail.com

Submitted on 05-Feb-2026

Abstract

Physics-Informed Neural Networks (PINNs) offer an effective framework for solving partial differential equations by incorporating governing physical laws into the learning process. When applied to quantum mechanical differential equations such as the Schrödinger equation, this approach produces smooth wavefunctions suitable for computing both conventional observables and Bohmian quantities. In this work, a PINNs based framework is developed to solve the one-dimensional time-dependent Schrödinger equation for the Quantum Harmonic Oscillator, a fundamental model with broad relevance in quantum and plasma physics. A fully connected neural network is used to represent the real and imaginary parts of the wavefunction, with the Schrödinger equation enforced through the loss function together with appropriate initial and boundary conditions. Automatic differentiation is employed to compute the required derivatives and to extract physical quantities from the learned solution. The method accurately reproduces the ground-state wavefunction and yields a ground-state energy in agreement with the analytical result. The Bohm quantum potential obtained from the network also matches the corresponding analytical expression, demonstrating that the essential spatial structure of the quantum amplitude is well captured. These results demonstrate that PINNs constitute a reliable and conceptually transparent framework for solving the time-dependent Schrödinger equation. Within this differential equation based setting, PINNs also enable systematic exploration of Bohmian aspects of quantum systems, where Bohmian quantities such as the quantum potential provide physical insight into confinement effects and the emergence of effective potentials.

1 Introduction

The time-dependent Schrödinger equation (TDSE) plays a central role in quantum mechanics, governing the dynamical evolution of quantum systems [1]. It governs the evolution of quantum systems within the framework of non-relativistic quantum mechanics. Considering a particle of mass m in an external potential $V(x)$, its dynamics are governed by the one-dimensional time-dependent Schrödinger equation

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right]\psi(x,t), \tag{1}$$

Here, $\psi(x,t)$ represents the system's complex wavefunction, and the probability of locating the particle at position x and time t is given by $|\psi(x,t)|^2$. Once the wavefunction is known, physical observables such as expectation values of position, momentum, and energy can be calculated [2]. The TDSE is required for describing a wide range of quantum phenomena, including wave-packet evolution, quantum tunneling, bound-state dynamics, and interactions with time-dependent external fields [3]. It provides a complete mathematical description of how quantum states change with time and is therefore essential in areas such as atomic and molecular physics, condensed matter physics, quantum optics, and plasma physics [4] . Accurate solutions of the TDSE are necessary for both theoretical investigations and practical applications involving quantum systems.

Only a limited number of idealized potentials admit analytical solutions of the TDSE, among them the free particle [5] , infinite well [6] , and harmonic oscillator [7]. Among these systems, the quantum harmonic oscillator (QHO) is of particular importance due to its wide applicability as an approximation for physical systems near equilibrium and its central role in quantum theory [8] . Analytical solutions of the harmonic oscillator provide closed-form expressions for energy eigenvalues and wavefunctions and serve as reliable reference results. Nevertheless, current research focuses not only on reproducing such solutions but also on developing methodologies that remain effective for systems lacking analytical solutions.

In this context, the QHO serves as an important benchmark problem. Its well-known exact solution allows direct validation of alternative computational approaches, enabling quantitative assessment of accuracy, stability, and physical consistency before extending these methods to more complex or analytically intractable systems.

To treat systems beyond analytical reach, several numerical methods have been developed for solving the TDSE, including finite-difference [9] , spectral and pseudo-spectral schemes [10] , split-operator techniques [11] , and implicit methods such as the Crank Nicolson approach [12] . Although traditional numerical methods have been successfully applied to many quantum

systems, they rely on explicit discretization of space and time using fixed grids. This dependence can increase computational cost, particularly for higher-dimensional problems, and may lead to numerical stability issues or the accumulation of errors during long-time evolution [13] . As a result, there is growing interest in alternative solution approaches that can provide smooth and accurate representations of quantum dynamics while reducing reliance on discretization.

To address these challenges, we employ Physics-Informed Neural Networks (PINNs) [14] , which allow us to approximate the wavefunction continuously over space and time without relying on fixed grids. PINNs incorporate the underlying physical laws, such as the Schrödinger equation, directly into the loss function, ensuring that the neural network solutions respect the governing quantum dynamics while enabling efficient and flexible computation [14] .

Until now, PINNs have been widely used to solve classical and quantum partial differential equations like the time-independent Schrödinger equation (TISE) and other evolution equations, where the neural network is trained to minimize the residuals of the governing equations across the computational domain [14] . By representing the solution as a neural network, we can evaluate it at any point in space and time, avoiding grid-based discretization and achieving a smooth and differentiable approximation. In this work, we extend the PINNs approach to solve the TDSE, enabling direct simulation of quantum dynamics and the evolution of wavefunctions in time, which is crucial for understanding dynamical processes in quantum systems.

Therefore, PINNs are employed in this work to solve the TDSE, using the QHO as a test system. The QHO is chosen not because its solution is unknown, but because it is a simple and well-understood system. Since its exact analytical solution is available, it provides a clear reference that allows new solution approaches to be tested and validated against known results before being extended to more complex quantum systems.

2 Methodology

To model the time evolution of quantum systems, we consider the TDSE. In this work, a one-dimensional QHO is selected as the physical system, where the external potential is well defined and analytically known. For an efficient neural-network-based treatment, the complex-valued wavefunction is decomposed into its real and imaginary parts, allowing the TDSE to be rewritten as a set of coupled real-valued partial differential equations. PINNs are then employed to approximate these components by embedding the governing equations together with the initial and boundary conditions directly into the learning framework.

The time evolution of the quantum system is described using the one-dimensional

TDSE .

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t), \tag{2}$$

where $\psi(x,t)$ is the complex-valued wavefunction and $V(x)$ denotes the external potential.

In this work, we consider the QHO potential,

$$V(x) = \frac{1}{2}m\omega^2x^2, \tag{3}$$

which serves as a simple and well-understood test system with an exact analytical solution.

For numerical convenience and improved training stability, the equation is expressed as dimensionless form by setting $\hbar = 1$ and $m = 1$. The TDSE then reduces to[15]

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{1}{2}\frac{\partial^2\psi(x,t)}{\partial x^2} + \frac{1}{2}\omega^2x^2\psi(x,t). \tag{4}$$

Since neural networks operate on real-valued functions, the complex wavefunction is decomposed as

$$\psi(x,t) = u(x,t) + i\,v(x,t), \tag{5}$$

where $u(x,t)$ and $v(x,t)$ are the real and imaginary components, respectively. Substituting this form into the TDSE yields the following coupled system of real-valued equations [13]:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial^2 v}{\partial x^2} - \frac{1}{2}\omega^2x^2v = 0, \tag{6}$$

$$\frac{\partial v}{\partial t} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\omega^2x^2u = 0. \tag{7}$$

The problem is defined within a finite space–time region.

$$x \in [-L, L], \quad t \in [0, T], \tag{8}$$

where L and T are chosen sufficiently large to ensure that boundary effects do not influence the solution within the region of interest. This bounded domain is required for the effective implementation of PINNs, as it enables stable evaluation of the governing equation residuals and consistent enforcement of physical constraints throughout the computational region.

Initial condition is

$$\psi(x,0) = \psi_0(x), \tag{9}$$

which corresponds to

$$u(x,0) = \Re[\psi_0(x)], \quad v(x,0) = \Im[\psi_0(x)]. \tag{10}$$

Homogeneous boundary conditions are imposed as

$$\psi(-L,t) = \psi(L,t) = 0, \tag{11}$$

ensuring confinement and numerical stability.

The Physics-Informed Neural Network approximates the functions $u(x,t)$ and $v(x,t)$ by minimizing a composite loss function defined as

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}}\,\mathcal{L}_{\text{IC}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}}, \tag{12}$$

where $\mathcal{L}_{\text{PDE}}$ enforces the TDSE residuals, while $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{BC}}$ impose the initial and

boundary conditions.The weighting parameters λ_{IC} and λ_{BC} regulate the relative importance of the initial and boundary condition constraints in the total loss function. Since different loss components may have different scales and convergence behaviors, these weights help balance the optimization process and prevent any single term from dominating the training [16] . Appropriate weighting improves numerical stability and convergence, and ensures that the learned solution simultaneously satisfies the governing differential equation and the prescribed physical conditions.
Within this bounded space-time domain, the solution of the governing equations is approximated using a Physics-Informed Neural Network. The PINNs framework enables the continuous representation of the wavefunction by embedding the TDSE and the associated constraints directly into the learning process.

To solve the governing equations within the defined space-time domain, we employ a framework which enables a mesh free solutions of partial differential equations by embedding physical constraints within the training process. Here, a PINNs is developed to model the real and imaginary parts of the quantum wavefunction over the full space–time domain.

The spatial and temporal coordinates (x, t) are used as inputs to the network, which outputs the real and imaginary components of the wavefunction, $u(x, t)$ and $v(x, t)$.Automatic differentiation is used to compute the required spatial and temporal derivatives, which are then substituted into the governing differential equations to evaluate the physics-based residuals. The total loss function is constructed by combining the differential equation loss with the losses corresponding to the initial and boundary conditions. By minimizing this total loss, the network parameters are optimized such that the predicted solution satisfies both the governing physical laws and the imposed constraints across the entire space-time domain.

2.1 Model Architecture and Training Setup

This section describes the network architecture and training configuration used in the PINNs framework. These choices determine the expressive capacity of the model, numerical stability, and convergence behavior during training.

The PINN employed in this work is implemented as a fully connected feedforward neural network designed to approximate the real and imaginary components of the quantum wavefunction.

- **Network architecture:** A fully connected feedforward neural network with three hidden layers, each containing 50 neurons.
- **Fully connected network:** A fully connected feedforward neural network is used, where each neuron is connected to all neurons in the next layer. This structure enables the learning of

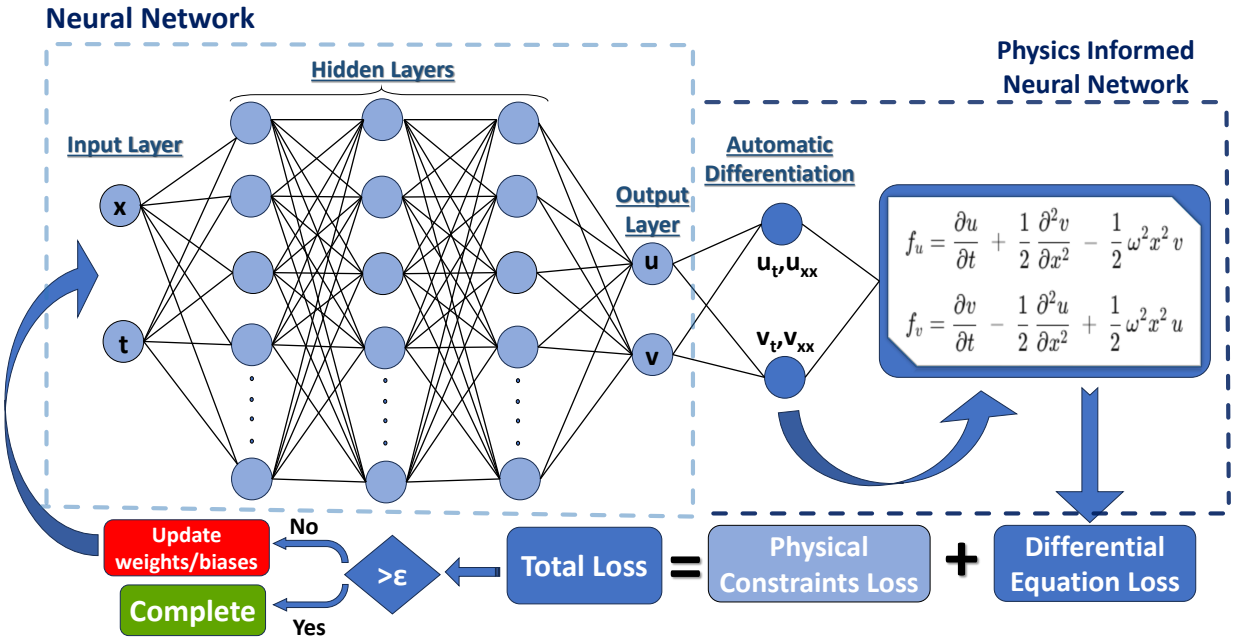


Figure 1: Architecture of a Physics-Informed Neural Network (PINNs). [17]

smooth, nonlinear mappings required for enforcing the governing physical equations within the PINNs framework [14].

- **Activation function:** The hyperbolic tangent (*tanh*) activation function is used in all hidden layers to ensure smooth approximations of the wavefunction and its derivatives.

- **Input variables:** Spatial and temporal coordinates (x, t).

- **Output variables:** The real and imaginary components of the wavefunction, denoted by $u(x,t)$ and $v(x,t)$, respectively.

- **Envelope function:** A Gaussian envelope $\exp(-x^2/2)$ is incorporated into the network output to enforce physical decay of the wavefunction at large spatial distances.

- **Training domain:** The bounded space time domain is defined as

$$x \in [-L, L], \quad L = 6, \qquad t \in [0, T], \quad T = 1,$$

.

- **Training points:** $N_f = 3000$ collocation points are randomly sampled within the interior of the domain to enforce the TDSE residual.
$N_{\text{IC}} = 900$ points are used to impose the initial condition at $t = 0$.

$N_{\mathrm{BC}} = 900$ points are sampled at $x = \pm L$ to enforce boundary conditions.

- **Optimizer:** The Adam optimizer is used for training.
- **Learning rate:** A constant learning rate of 10^{-3} is employed.
- **Training epochs:** The network is trained for a total of 5000 epochs.
- **Automatic differentiation:** All spatial and temporal derivatives required for the loss function are computed using automatic differentiation.

2.2 Loss Function Formulation

The PINN is trained by minimizing a composite loss function constructed from physically motivated terms that enforce the governing time-dependent Schrödinger equation (TDSE), as well as the associated initial and boundary conditions. The total loss function is defined as

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{TDSE}} + \mathcal{L}_{\mathrm{IC}} + \lambda_{\mathrm{BC}}\,\mathcal{L}_{\mathrm{BC}} + \lambda_{\mathrm{phase}}\,\mathcal{L}_{\mathrm{phase}}, \tag{13}$$

where λ_{BC} and λ_{phase} are weighting parameters that balance the relative contributions of the boundary and phase-consistency constraints.

- **TDSE residual loss:** Enforces the governing TDSE within the interior of the space–time domain [18]. It is defined as

$$\mathcal{L}_{\mathrm{TDSE}} = \frac{1}{N_f}\sum_{j=1}^{N_f}\left[f_u(x_j,t_j)^2 + f_v(x_j,t_j)^2\right], \tag{14}$$

where f_u and f_v denote the residuals of the coupled real and imaginary TDSE equations [19].

- **Initial condition loss:** Enforces agreement between the network prediction and the prescribed initial wavefunction at $t = 0$.
- **Boundary condition loss:** Enforces physical decay of the wavefunction at the spatial boundaries by constraining both the wavefunction values and their spatial derivatives,

$$\mathcal{L}_{\mathrm{BC}} = \left\langle u^2 + v^2 \right\rangle_{\pm L} + \left\langle (\partial_x u)^2 + (\partial_x v)^2 \right\rangle_{\pm L}. \tag{15}$$

- **Phase-consistency loss:** Preserves the correct eigenstate time evolution of the QHO,

$$\mathcal{L}_{\mathrm{phase}} = \left\langle \left(u_t + E_0 v\right)^2 + \left(v_t - E_0 u\right)^2 \right\rangle, \tag{16}$$

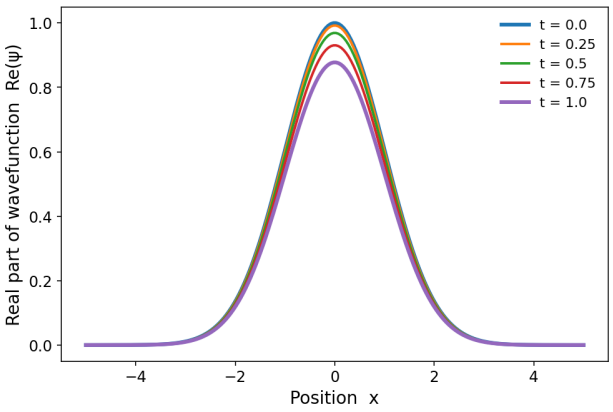
where $E_0 = \frac{1}{2}\omega$ is the ground-state energy.

These loss components collectively ensure accurate enforcement of the governing physics, numerical stability during training, and reliable approximation of the wavefunction over the entire space–time domain.

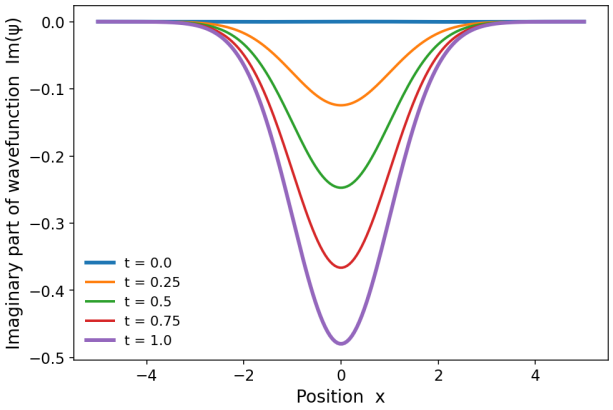
3 Result and Discussion

The Physics-Informed Neural Network (PINNs) accurately solves the TDSE for the one-dimensional QHO. The network is trained by enforcing the governing

equation, the ground-state initial condition, boundary constraints, and a phase-consistency condition to ensure correct temporal evolution.



(a)



(b)

Figure 2: Time evolution of the real (a) and imaginary (b) components of the wavefunction.

The real and imaginary components of the wavefunction in Fig. 2a and 2b demonstrate the correct time-dependent evolution. The wavefunction is purely real at the initial time, while the imaginary component develops smoothly as time progresses. The two components oscillate in time with a constant phase difference, demonstrating that the network preserves the correct quantum-mechanical phase structure. The use of a Gaussian envelope in the network architecture enforces proper spatial decay, while the imposed boundary constraints ensure numerical stability and prevent non-physical behavior near the domain boundaries. Overall, these results validate the effectiveness and physical consistency of the proposed PINNs framework for time-dependent quantum systems.

Using the learned wavefunction obtained from the PINNs solution, we further construct the Bohm quantum potential [20] , which provides an alternative description of quantum dynamics within the Bohmian formulation of quantum mechanics. In this approach, quantum effects are encoded through an additional potential term derived entirely from the wavefunction amplitude.

The complex wavefunction is written in polar form as

$$\psi(x,t) = R(x,t)\, e^{iS(x,t)},$$

where the amplitude $R(x,t)$ and phase $S(x,t)$ are real-valued functions. From the PINNs-predicted real and imaginary components,

$$\psi(x,t) = u(x,t) + i\, v(x,t),$$

the amplitude is computed as [21]

$$R(x,t) = \sqrt{u(x,t)^2 + v(x,t)^2}.$$

The Bohm quantum potential is defined as

$$V_{\text{Bohm}}(x,t) = -\frac{1}{2}\frac{\nabla^2 R(x,t)}{R(x,t)},$$

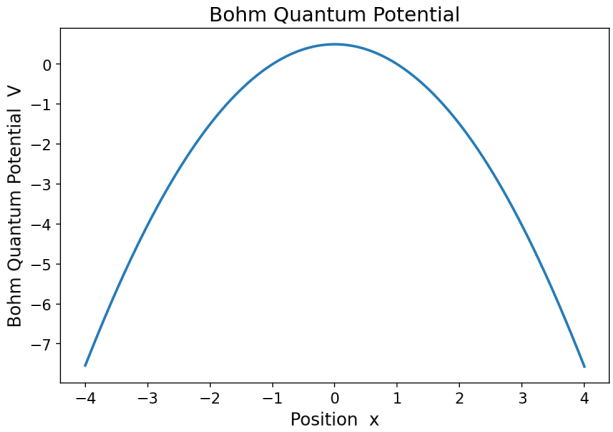
which, for the present one-dimensional system, reduces to

$$V_{\text{Bohm}}(x,t) = -\frac{1}{2}\frac{\partial^2 R(x,t)}{\partial x^2} \Big/ R(x,t).$$

All spatial derivatives required for the evaluation of the Bohm potential are computed using automatic differentiation [22], ensuring a smooth and accurate calculation without numerical discretization errors. The resulting Bohm potential in Fig. 3a exhibits a symmetric and well-behaved spatial profile, consistent with the expected structure QHO.

The Bohm quantum potential obtained in this work is direct relevance to quantum plasma physics, where it represents the leading-order quantum correction to classical fluid models [23] . In quantum hydrodynamic descriptions of plasmas, the Bohm potential accounts for wavefunction spreading and tunneling effects that arise from the quantum nature of charged particles [24].

The smooth and physically consistent Bohm potential constructed from the PINNs-predicted wavefunction demonstrates that the present work can accurately resolve quantum effects that are essential in dense or low-temperature plasmas. An important advantage of the PINNs-based approach is the accurate evaluation of the Bohm potential without numerical noise, since all spatial derivatives are computed using automatic differentiation. This feature



(a)

Figure 3: Bohm quantum potential corresponding to the ground-state wavefunction.

is particularly valuable for plasma applications, where small errors in the Bohm term can lead to unphysical instabilities in quantum hydrodynamic simulations.

These results indicate that the methodology developed here can be readily extended to quantum plasma models, providing a reliable tool for incorporating quantum effects into plasma simulations while avoiding the limitations of traditional grid-based numerical schemes.

4 Conclusion

In this study, the TDSE for a one-dimensional QHO was solved using a Physics-Informed Neural Network approach. The governing equation was embedded directly into the loss function along with suitable initial and boundary conditions, allowing the wavefunction to be learned without relying on traditional nu-

merical discretization methods. The QHO was chosen as a test system because its exact solutions are well known. The PINNs accurately capture the time evolution of both the real and imaginary components of the wavefunction. The extracted ground-state energy agrees closely with the expected analytical value, confirming the correctness of the model and training strategy.

An important outcome of this work is the construction of the Bohm quantum potential from the learned wavefunction. Using spatial derivatives obtained via automatic differentiation, the Bohm potential was calculated and found to match the expected behavior for the harmonic oscillator. This demonstrates that the PINNs solution is not only accurate at the wavefunction level but also reliable for computing derived physical quantities. Overall, the results show that PINNs provide a robust and a versatile framework for solving the time-dependent Schrödinger equation and analyzing quantum systems. The methodology presented here can be extended to more complicated potentials and boundary conditions, where analytical solutions are not available.

Acknowledgments

A. Awasthi acknowledges the financial support provided by the DST, Govt. of India, through Grant No. DST/INSPIRE Fellowship/2020/IF200538.

References

[1] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. 2nd ed. Cambridge: Cambridge University Press, 2017.

[2] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. 3rd ed. Cambridge: Cambridge University Press, 2018.

[3] N. Sathyamurthy and S. Mahapatra. "Time-dependent quantum mechanical wave packet dynamics". In: *Phys. Chem. Chem. Phys.* 23 (2021), pp. 7586–7614. DOI: 10.1039/D0CP03929B.

[4] Abraham Nitzan. "Quantum dynamics using the time-dependent Schrödinger equation". In: *Chemical Dynamics in Condensed Phases: Relaxation, Transfer, and Reactions in Condensed Molecular Systems*. Ed. by Abraham Nitzan. Oxford University Press, 2024, pp. 56–112. DOI: 10.1093/9780191947971.003.0002. URL: https://academic.oup.com/book/57505/chapter/467230208.

[5] R. W. Robinett. "Analytic time-dependent solutions of the one-dimensional Schrödinger equation". In: *American Journal of Physics* 82.10 (2014), pp. 955–962. DOI: 10.1119/1.4894549.

[6] M. S. Perepelkin, G. V. Sakhno, and I. N. Toptygin. "Exact time-dependent solution of the Schrödinger equation

and its generalization to phase space". In: *arXiv preprint* (2021). arXiv: 2112.15212 [quant-ph].

[7] A. C. J. Coelho, F. P. dos Santos, and A. de Souza Dutra. "Exact Solution of a Time-Dependent Quantum Harmonic Oscillator". In: *Entropy* 24.12 (2022), p. 1851. DOI: 10.3390/e24121851.

[8] A. Mahmoud. "The Quantum Harmonic Oscillator: Mathematical Framework and Applications in Modern Physics". In: *ResearchGate Preprint* (2025). URL: https://www.researchgate.net/publication/395695112.

[9] R. Becerril, L. Vázquez, and J. Caballero. "Solving the time-dependent Schrödinger equation using finite difference methods". In: *Revista Mexicana de Física E* 54.2 (2008), pp. 120–126.

[10] Yuya Suzuki, Frances Y. Kuo, and Ian H. Sloan. "Strang splitting in combination with rank-1 and rank-r lattices for the time-dependent Schrödinger equation". In: *arXiv preprint* (2018). arXiv: 1808.06357 [math.NA].

[11] N. Moiseyev, M. Šindelka, and L. S. Cederbaum. "Numerical solution of the three-dimensional time-dependent Schrödinger equation in spherical coordinates: Spectral basis and effects of split-operator technique". In: *Journal of Computational and Applied Mathematics* 225.1 (2009), pp. 56–67. DOI: 10.1016/j.cam.2008.06.015.

[12] O. S. K. Sastri. *Computer Simulations in Quantum Physics: Using Numerical Techniques, Worksheets and Programming*. Available at Amazon India. India: Notion Press, 2023. ISBN: 978-935206XXXX. URL: https://www.amazon.in/Computer-Simulations-Quantum-Physics-Programming/dp/B0CMXVW4PV.

[13] Ryan Schneider and Heman Gharibnejad. "Numerical Methods for the Time-Dependent Schrödinger Equation: Beyond Short-Time Propagators". In: *Atoms* 13.8 (2025), p. 70. DOI: 10.3390/atoms13080070.

[14] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[15] F. M. Fernández. "Dimensionless equations in non-relativistic quantum mechanics". In: *arXiv preprint* arXiv:2005.05377 (2020). Available at https://arxiv.org/abs/2005.05377.

[16] Zixue Xiang et al. "Self-adaptive loss balanced Physics-informed neural networks". In: *Neurocomputing* 496

(2022), pp. 11–34. DOI: 10.1016/j.neucom.2022.05.015.

[17] European Space Agency (ESA). *Advanced Concepts Team – About the Team*. https://www.esa.int/gsp/ACT/about/theteam. Accessed: 2025-02-04. 2025.

[18] Zijian Zhou and Zhen-Ya Yan. "Deep learning neural networks for the third-order nonlinear Schrödinger equation: Bright solitons". In: *Physics Letters A* 381.20 (2017), pp. 1737–1743. DOI: 10.1016/j.physleta.2017.03.039.

[19] Karan Shah et al. "Physics-Informed Neural Networks as Solvers for the Time-Dependent Schrödinger Equation". In: *arXiv:2210.12522 [quant-ph]* (2022). Preprint, arXiv. URL: https://doi.org/10.48550/arXiv.2210.12522.

[20] A. Benseny et al. "Applied Bohmian Mechanics". In: *arXiv preprint* (2014). arXiv: 1406.3151 [quant-ph].

[21] David Bohm et al. *Bohmian Mechanics*. https://plato.stanford.edu/entries/qm-bohm/. Stanford Encyclopedia of Philosophy, edited by Edward N. Zalta. 2020. (Visited on 02/04/2026).

[22] Chuqi Chen et al. "Automatic Differentiation is Essential in Training Neural Networks for Solving Differential Equations". In: *Journal of Scientific Computing* 104 (2025), p. 54. DOI: 10.1007/s10915-025-02965-3.

[23] D. Michta, F. Graziani, and M. Bonitz. "Quantum hydrodynamics for plasmas—a Thomas–Fermi theory perspective". In: *Contributions to Plasma Physics* 55.5–6 (2015), pp. 437–447. DOI: 10.1002/ctpp.201500042. arXiv: 1503.04037 [physics.plasm-ph].

[24] A. S. Sanz and S. Miret-Artés. "Setting up tunneling conditions by means of Bohmian mechanics". In: *Journal of Physics A: Mathematical and Theoretical* 44.48 (2011), p. 485301. DOI: 10.1088/1751-8113/44/48/485301.

Physics Informed Neural Networks (PINNs) Approach for Numerical Solution of Heat Equation

Gargi Rathore[1], Arushi Sharma[1] and Ishwar Kant[1]

[1] Department of Physics and Astronomical Science
Central University of Himachal Pradesh, Dharamshala, Bharat.

Submitted on 05-Feb-2026

Abstract

Heat conduction is a fundamental physical process governing thermal energy transport across diverse media. While traditional numerical methods (e.g., Finite Difference) are physically powerful, they often require complex and can be computationally expensive for high dimension. PINNs utilize the universal approximation power of neural networks constrained by the underlying physical laws. We present a Python-based implementation demonstrating how PINNs can be used to visualize thermal diffusion. A physics-informed neural networks is formulated by embedding the governing heat equation, along with the prescribed initial and boundary conditions, into the loss function of a neural network. Automatic differentiation is employed to compute the required spatial and temporal derivatives. The network is trained using collocation points sampled across the space–time domain, without the need for labeled training data. The PINNs-based solution exhibits excellent agreement with the actual solution of the heat equation, correctly reproducing the spatial temperature profiles and their temporal evolution. The study highlights the effectiveness of PINNs as a reliable alternative to traditional numerical methods.

1 The Heat Equation

The heat equation is a very special case of the general linear second-order partial differential equation[3]. The heat equation is a parabolic partial differential equation and occurs in the characterization of diffusion progress. The one-dimensional heat equation is:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{1}$$

where $u = u(x,t)$ is the dependent variable, and α is a constant coefficient. Equation 1 is a model of transient heat conduction in a slab of material with thickness L. The solution is defined on a strip of width L that extends infinitely in time. The material property α is the thermal diffusivity. In a practical computation, the solution is obtained only for a finite time, say t_{max}. Solution to Equation 1 requires specification of bound-

ary conditions at x =0 and x =L, and initial conditions at t = 0. In numerical calculations, the solution is evaluated only up to a finite time t_{max}. Equation 1 must be supplemented with boundary conditions at x =0 and x =L, along with an initial condition at t=0.

Simple boundary and initial conditions are

$u(0,t) = u_0, \quad u(L,t) = u_L \quad u(x,0) = f_0(x).$

Other boundary conditions, e.g. gradient (Neumann) or mixed conditions or dritchlet condition, can be specified.

For a homogeneous and isotropic medium where the material properties are uniform and identical in all directions and assuming constant thermal properties, the 2D heat conduction equation is expressed as

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{2}$$

where

- The function $u(x,y,t)$ denotes the temperature at position (x,y) at time t. It describes the thermal state of the medium and its temporal evolution, showing how the temperature varies due to internal and external influences.
- The parameter α represents the thermal diffusivity of the medium, which characterizes the rate at which heat spreads through the material.
- $\frac{\partial u}{\partial t}$ signifies the rate of change of temperature with respect to time, indicating the dynamic evolution of the thermal state. This term is essential for understanding transient heat conduction, where temperature changes occur over time.
- $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ are the second-order partial derivatives of temperature with respect to the spatial coordinates x and y, respectively. They characterize the local curvature of the temperature profile and indicate how variations in spatial gradients govern heat transport in the medium.

These equations serves as a mathematical model to describe the temporal and spatial distribution of temperature in a two-dimensional medium. It is crucial for understanding how heat propagates within materials and across various mediums, impacting a wide range of applications from industrial processes to environmental science.

Traditional numerical approaches, such as the finite difference and finite element methods, are based on discretizing the governing physical equations. In the finite difference method, derivatives in the differential equation are replaced by appropriate difference approximations[5]. For the heat equation, this involves discretization in both time and space. Different choices of grid points in these approximations lead to different numerical methods, each with its own convergence behavior. The rate at which the numerical solution converges to the exact solution depends on the method employed. Moreover, certain practical formulations may become unstable for inappropri-

ate choices of the spatial and temporal step sizes, Δx and Δt.

This work investigates the simulation of one- and two-dimensional heat conduction problems using Physics-Informed Neural Networks. The objective is to develop a robust PINN-based framework for modeling one- and two-dimensional heat conduction with boundary conditions, while exploiting neural networks to embed the underlying physical laws into the learning process.[6, 7]

Assume that we have a thermal system $u(x,t)$, where x and t stand for the spatial and temporal coordinates, respectively. To minimize the residual of the heat equation, $\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0$, a neural network is trained using the PINNs approach[6]. Even in 2D environments where conventional grid-based approaches would necessitate complicated indexing, using PINNs, this results in a highly accurate global approximation of the temperature field[6, 7, 2].

2 Physics Informed Neural Networks(PINNs)

In a traditional neural network, the model functions as a universal functional approximator, mapping inputs to outputs through a purely data-driven regime [1, 2]. Traditional Neural Network (TNN) learns patterns solely from labeled datasets without any intrinsic knowledge of the underlying governing equations. Three different kinds of layers make up the architecture:

- **Input Layer:** This layer gets the raw coordinate data, like time (t) and space (x, y). The number of neurons corresponds directly to the dimensionality of the input space.
- **Hidden Layers:** Non-linear transformations are carried out by these intermediate layers. Each layer consists of "neurons" that extract features from the input data. To balance computational efficiency and approximation power, we employ 64-neuron hidden layers.
- **Output Layer:** $\hat{u}$, is the temperature generated by the network's output layer.

In order to minimize the difference between predictions and observations, the network learns by modifying its internal parameters:

- **Weights and Biases:** Every connection between neurons has an associated weight W that determines how strongly one neuron influences another. In addition, each neuron has a bias b, which is a trainable constant added to the weighted sum of inputs to allow the neuron to shift its activation. These parameters are adjusted during training .

$$\mathbf{a} = \sigma\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right) \tag{3}$$

Where:

  - $\mathbf{x}$ is the input coordinate vector.
  - $\mathbf{W}$ is the weight matrix.
  - $\mathbf{b}$ is the bias vector.

  - σ is the non-linear activation function.
  - **a** is the output activation of the layer.

- **Activation Function:** The hyperbolic tangent (*tanh*) function is utilized. Because physical problems often require computing second-order derivatives, a smooth, infinitely differentiable activation function is necessary.[7]

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

- **Loss Function:** The NN model is trained by minimizing the Mean Squared Error (MSE):

$$\mathcal{L}_{data} = \frac{1}{N}\sum_{i=1}^{N} |\hat{u}_i - u_i|^2 \tag{4}$$

The Traditional Neural Network does not ensure physical consistency between training points because the NN objective is restricted to minimizing $\mathcal{L}_{data}$.

PINNs, on the other hand, follows the rules of physics by minimizes a composite loss function. [6]. This is made possible by Automatic Differentiation (AD), which enables the network to precisely compute its own gradients[7].

Over a time interval $[0, t_{max}]$, the field $u(\mathbf{x}, t)$ in a 2D spatial domain Ω changes in accordance with:

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0, t_{max}] \tag{5}$$

where the thermal diffusivity is represented by α. The system's initial state and boundary behavior must be specified in order to produce a unique solution[3].

- **Initial Condition (IC):** The state of the system at $t = 0$.

$$u(\mathbf{x}, 0) = h(\mathbf{x}) \tag{6}$$

where $h(\mathbf{x})$ usually denotes a particular distribution or a localized heat source.

- **Boundary Conditions (BC):** When the field value is fixed at the domain edges $\partial\Omega$, we use Dirichlet boundary conditions.

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega \tag{7}$$

The sum of the Mean Squared Errors (MSE) from three different sources is the total objective function $\mathcal{L}$:

$$\mathcal{L}_{Total} = \omega_f \mathcal{L}_f + \omega_{bc} \mathcal{L}_{bc} + \omega_{ic} \mathcal{L}_{ic} \tag{8}$$

Where:

- **Physics Residual Loss($\mathcal{L}_f$):**
  - **1-D Case:**

$$\mathcal{L}_f = \frac{1}{N_f}\sum_{i=1}^{N_f} \left| \frac{\partial \hat{u}}{\partial t} - \alpha \frac{\partial^2 \hat{u}}{\partial x^2} \right|^2 \tag{9}$$

  - **2-D Case:**

$$\mathcal{L}_f = \frac{1}{N_f}\sum_{i=1}^{N_f} \left| \frac{\partial \hat{u}}{\partial t} - \alpha \left(\frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} \right) \right|^2 \tag{10}$$

- **Boundary Condition Loss ($\mathcal{L}_{bc}$):**

$$\mathcal{L}_{bc} = \frac{1}{N_{bc}}\sum_{j=1}^{N_{bc}} |\hat{u} - g|^2 \tag{11}$$

- **Initial Condition Loss ($\mathcal{L}_{ic}$):**

$$\mathcal{L}_{ic} = \frac{1}{N_{ic}} \sum_{k=1}^{N_{ic}} |\hat{u} - h|^2 \qquad (12)$$

Nevertheless, the method is susceptible to Spectral Bias, in which the network learns low-frequency, smooth patterns more quickly than acute thermal gradients[2]. Our use of 64-neuron hidden layers and a learning rate of 10^{-3} helps to alleviate this, guaranteeing a balance between detail and stability[7].

3 Computational Implementation

Five different computational phases are used to solve the Heat Equation using Physics-Informed Neural Networks (PINNs), moving from raw domain sampling to a trained continuous solution manifold[6, 2].

3.1 Computational Procedure

Step 1: Point Sampling and Domain Initialization The continuous (x, t) domain is discretized into point sets prior to training:

- **Collocation Points:** A uniform random distribution is used to select 1,000 points throughout the domain in order to assess the PDE residual.
- **Constraint Points:** For the Initial Condition (IC), 200 points are fixed at $t = 0$, and for the Boundary Conditions (BC), 200 points are fixed at $x = 0$ and $x = 1$.

Step 2: Neural Network Architecture Setup
The initialization of the model $u(x, t) \approx NN(x, t; \theta)$ is as follows[1, 7]:

- **Input:** Input layer for coordinates (x, t) are connected to the hidden layers, each with 64 neurons.
- **Activation Function:** *tanh* functions to guarantee the heat equation's smooth, twice-differentiable curvature[7].
- **Output:** The temperature u is predicted by the output layer of the network.

Step 3: Physics-Informed Loss Construction
A composite loss function $\mathcal{L}_{total} = \mathcal{L}_f + \lambda(\mathcal{L}_i + \mathcal{L}_{bc})$ directs the "Learning": enforces the PDE using automatic differentiation: $u_t - ku_{xx} = 0$.

- **Residual Loss ($\mathcal{L}_f$):** Enforces the PDE $u_t - ku_{xx} = 0$ via Automatic Differentiation.
- **Initial Loss ($\mathcal{L}_i$):** Enforces $u(x, 0) = 0$.
- **Boundary Loss ($\mathcal{L}_{bc}$):** Enforces $u(0, t) = 1.0$ and $u(1, t) = 0.0$.

Step 4: Iterative Optimization
It consists of these two steps:

- **Optimizer:** It uses the **Adam** algorithm for stochastic weight updates[1].
- **Weighting:** To guarantee boundary convergence, $\mathcal{L}_i$ and $\mathcal{L}_{bc}$ are given a penalty factor ($\lambda = 10$). To achieve convergence, the procedure is iterated 4,000 times.

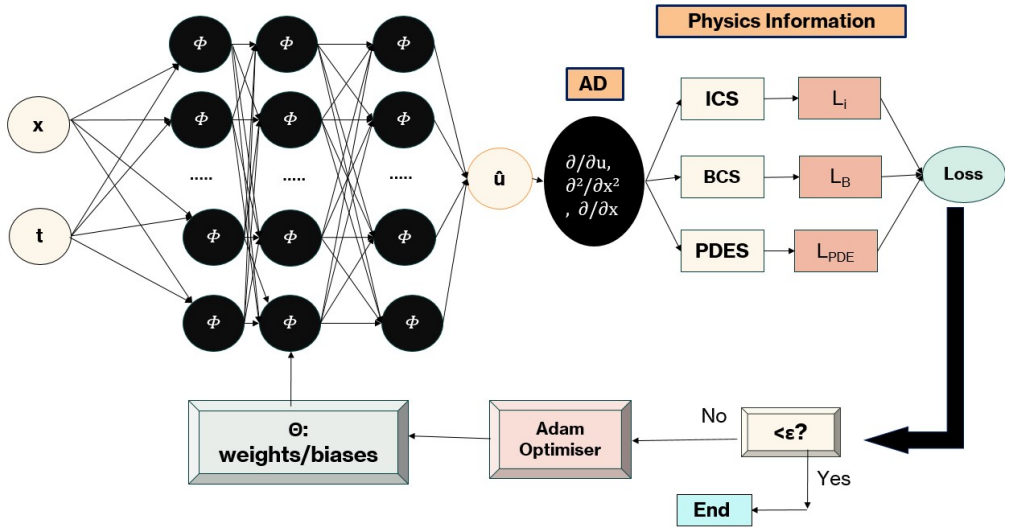


Figure 1: Architecture of a Physics-Informed Neural Network (PINN)

4 1D Heat Equation Results: Rod with Single-End Heater

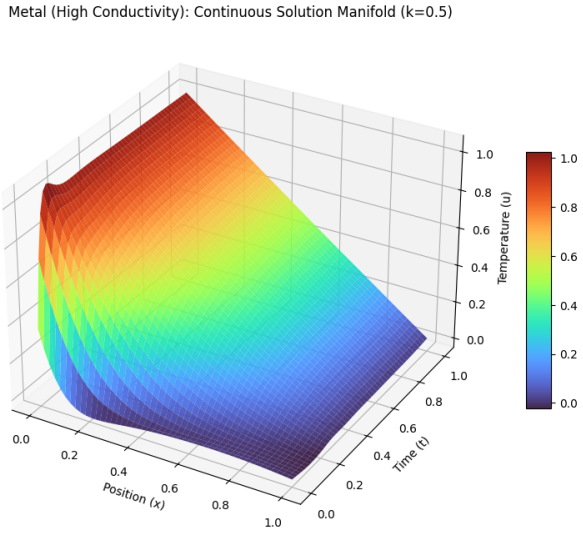
With a constant heater connected to the boundary at $x = 0$, the 1D simulation simulated a rod of length $L = 1.0$, keeping the temperature at $u(0,t) = 1.0$ for all $t > 0$. $u(x,0) = 0$ was the initial condition across the remainder of the rod. The continuous solution manifold over a temporal domain of $t \in [0,1]$ has to be resolved using the PINNs.

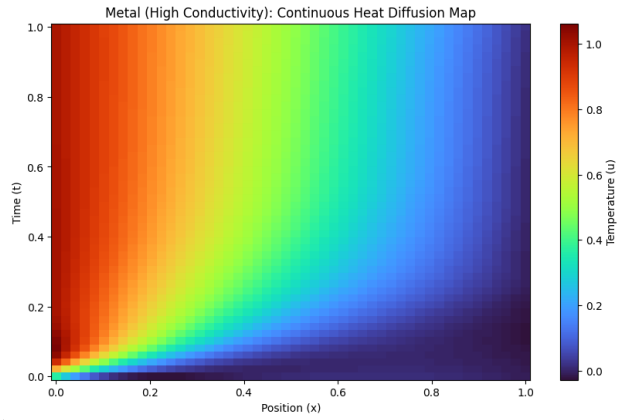
4.0.1 High-Conductivity Material (Metal)

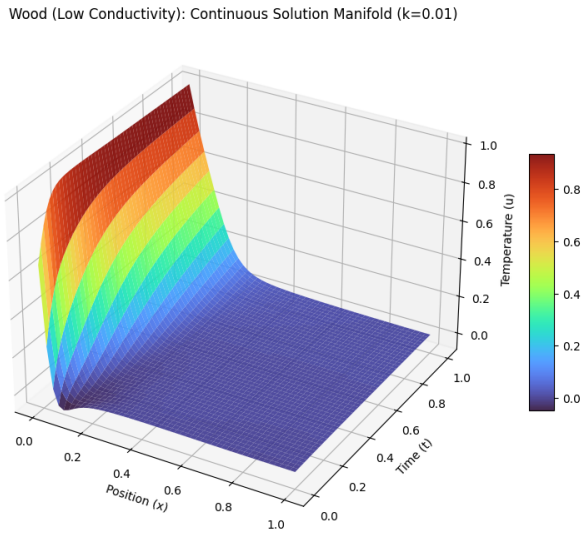
Rapid thermal propagation across the spatial domain was successfully captured by the PINNs for the metal simulation (set up with $k = 0.5$). As seen in the 3D solution manifold and the continuous heat diffusion map:

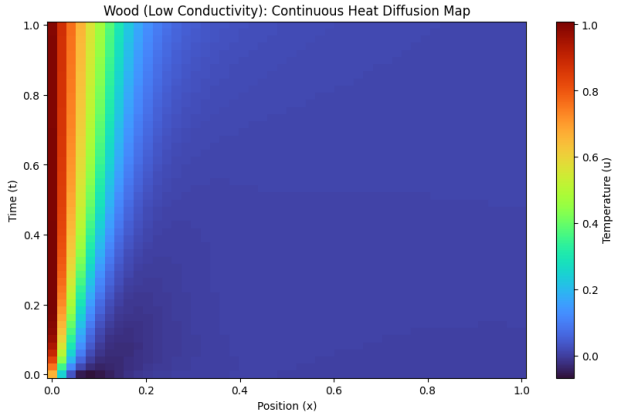
Efficient energy transfer is shown by the temperature profile's sweeping, smooth curvature. Even at the furthest points, the heater's thermal influence has reached the distal end of the rod ($x = 1.0$) by $t = 1.0$, with temperatures rising noticeably above the initial state.

Heatmap: Metal









4.0.2 Low-Conductivity Material (Wood)

The hardwood log (designed with $k = 0.01$) exhibits high localization and thermal resistance. The temperature increase is firmly limited to the immediate area of the heater ($x < 0.2$), according to the heat diffusion map. Near the source, the temperature rise but the rod stays at its starting ambient temperature for most of the time after $x = 0.2$. The PINNs remained stable and faithfully depicted the insulating behavior of the wood in spite of the steep gradients that usually provide a hurdle to conventional numerical solvers[7].

Heatmap: Wood

5 2D Heat Conduction in a Plate

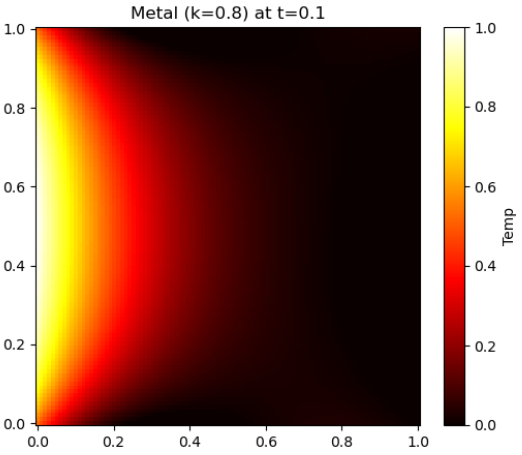
The simulation was extended to a 2D square plate of unit dimensions. The left edge ($x = 0$) was subjected to a continual heat source ($T = 1.0$), while the other limits were kept at $T = 0$. For two different materials, the PINNs successfully mapped the spatial-temporal evolution of heat.

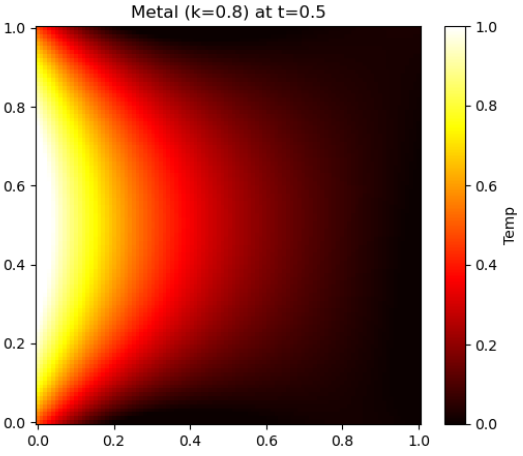
5.0.1 High-Conductivity Material (Metal)

The PINNs recorded a quick and wide thermal expansion, as seen in the heatmaps for

Metal ($k = 0.8$). The heat front had already made considerable inroads into the plate's center at $t = 0.1$. The efficient energy transfer feature of metallic structures is demonstrated by the thermal gradient being significantly smoother at $t = 0.5$, with the 0.2 temperature contour extending past the midpoint ($x = 0.6$).

Heatmap: Metal

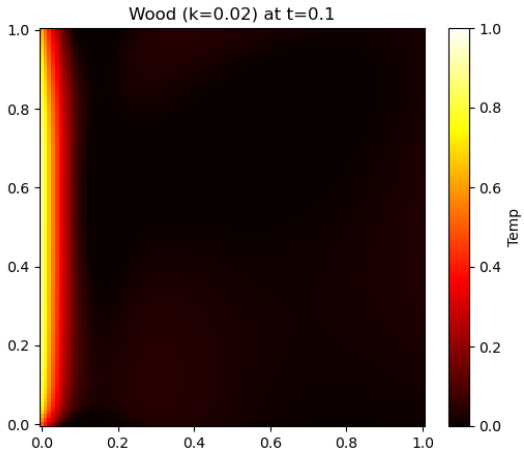


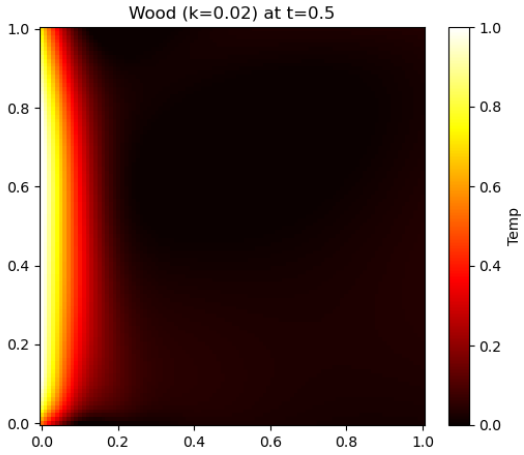


5.0.2 Low-Conductivity Material (Wood)

The Wood ($k = 0.02$) simulation, on the other hand, showed a very localized temperature response. The heat is still mostly contained in the area around the heated edge ($x < 0.2$) at both $t = 0.1$ and $t = 0.5$. The PINNs ability to resolve steep gradients in insulating materials without numerical instability is demonstrated by the abrupt change from the heater temperature to the ambient temperature.

Heatmap: Wood





The findings show that isotherms dispersed evenly over the domain and that the **metal plate** rapidly achieved a quasi-steady state. The PINNS correctly predicted a large thermal lag in the **wooden plate** simulation;

Table 1: Material Parameters used in 2D PINNs Simulations

Property	Metal	Wood
Conductivity (k)	0.8	0.02
BC ($x = 0$)	$T = 1.0$	$T = 1.0$
IC ($t = 0$)	$T = 0.0$	$T = 0.0$

during the simulation, the heat stayed focused along the $x = 0$ axis.

6 Conclusion

In this paper, we systematically show how well Physics-Informed Neural Networks (PINNs) solve the one- and two-dimensional heat equation. By incorporating the governing heat-conduction equation directly into the neural network loss function, the suggested PINN framework effectively captures the spatiotemporal evolution of the temperature field without the need for either temporal or spatial discretization. Compared to traditional numerical solvers like the Finite Difference Method (FDM) and the Finite Element Method (FEM), the PINN approach is more adaptable when dealing with complicated boundary and initial conditions. The findings show that PINNs offer a reliable and effective substitute for simulating heat conduction issues, especially when conventional grid-based approaches become computationally costly or challenging to use.

References

[1] Yoshua Bengio, Ian Goodfellow, Aaron Courville, et al. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[2] Miaomiao Chen, Ruiping Niu, and Wen Zheng. Adaptive multi-scale neural network with resnet blocks for solving partial differential equations. *Nonlinear Dynamics*, 111(7):6499–6518, 2023.

[3] Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.

[4] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[5] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[6] M Raissi, P Perdikaris, GE Karniadakis, and Bhavesh Shrimali. Cs598: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear pdes. 2021.

[7] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[8] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

Learning Solutions of Coupled Differential Equations with Physics-Informed Neural Networks

Aman Sharma[1], Ayushi Awasthi[1] and O.S.K.S. Sastri[1]

[1]Department of Physics and Astronomical Science,
Central University of Himachal Pradesh,
Dharamshala 176215, India.
rattanaman7@gmail.com

Submitted on 05-Feb-2026

Abstract

This study demonstrates the application of Physics-informed neural networks (PINNs) to solve the initial-value problem of a two-mass coupled nonlinear oscillator system. The system is governed by the coupled second-order ODEs.

A fully-connected feed-forward neural network is trained to directly approximate the displacement fields $x_1(t)$ and $x_2(t)$. The network is optimized by minimizing a composite loss function consisting of the physics residual and a weighted initial-condition loss. No labeled trajectory data are used to train the model. The PINNs solution is validated against a high-accuracy numerical method. The predicted displacements show excellent quantitative agreement with the numerical solution, achieving sub-millimeter root-mean-square errors for both masses throughout the simulated interval. These results illustrate that PINNs can accurately predict the solutions of nonlinear differential equations.

Keywords: Physics-Informed Neural Networks, Coupled Oscillators, Nonlinear Dynamics, Automatic Differentiation.

1 Introduction

Coupled oscillator systems are fundamental models in classical mechanics, appearing in a wide range of physical phenomena such as molecular vibrations, acoustic meta materials, electrical circuits, and mechanical structures with multiple degrees of freedom. When the restoring forces are linear, the system can be solved analytically using normal mode analysis. However in real world, most of systems exhibit nonlinearities such as cubic stiffness terms that lead to complex behaviors including amplitude-dependent frequencies, internal resonance, and energy transfer between modes. These

nonlinear effects are difficult to treat exactly and usually require numerical methods that become computationally expensive for long-time simulations or when many parameters must be explored.

Physics-informed neural networks (PINNs) have emerged as a best alternative for solving both forward and inverse problems governed by differential equations[1]. By embedding the governing equations directly into the loss function of a neural network via automatic differentiation, PINNs can approximate solutions without requiring any labeled simulation data. It requires training data in the form of differential equations. Since their introduction by Raissi et al. [2], PINNs have been successfully applied to a variety of problems including quantum mechanics [3], fluid dynamics & structural dynamics. Their ability to handle strong nonlinearities with relatively small networks makes them particularly attractive for nonlinear dynamical systems.

In this work we apply PINNs to the classic two-mass, three-spring system with an additional cubic nonlinear stiffness term on each mass. We demonstrate that a compact feed-forward neural network is trained solely on the physics residual and initial conditions. It can accurately reproduce the nonlinear beating phenomenon observed in the system over a long time interval. The PINNs solution is validated against a high-accuracy numerical solution obtained with SciPy's `odeint` solver, showing excellent agreement.

This study serves as a clear, reproducible demonstration that PINNs can capture the essential nonlinear dynamics of coupled oscillators with minimal computational overhead and no need for mesh generation or extensive training data. The results highlight the potential of PINNs as a versatile tool for exploring nonlinear mechanical systems, parameter studies, and real-time digital twins.

2 Methodology

In this section, we will explain the working of feed-forward neural network combined with physics informed loss functions shown in figure 1

2.1 Governing Equations

The system under consideration is a two-mass coupled nonlinear oscillator with cubic stiffness nonlinearity. The governing differential equations are

$$m\ddot{x}_1 + k_1 x_1 - k_c(x_2 - x_1) + k_{nl} x_1^3 = 0 \quad (1)$$

$$m\ddot{x}_2 + k_2 x_2 - k_c(x_1 - x_2) + k_{nl} x_2^3 = 0 \quad (2)$$

with symmetric linear stiffness ($k_1 = k_2 = 1$), weak coupling ($k_c = 0.5$), and a small nonlinear coefficient ($k_{nl} = 0.01$). Initial conditions are chosen to excite primarily the first mass: $x_1(0) = 1$, $x_2(0) = 0$, $\dot{x}_1(0) = \dot{x}_2(0) = 0$ [4].

2.2 Physics-Informed Neural Network

A fully connected feedforward neural network [5] in combination with physics informed loss functions was used to approximate the solution pair $(x_1(t), x_2(t))$. The network takes time t as the sole input and outputs a two-dimensional vector. It consists of an input layer with one neuron, three hidden layers each containing 64 neurons with hyperbolic tangent (tanh) activation functions, and an output layer with two neurons.

The neural network is trained iteratively using the following six-step process:

1. **Random Initialization:** At the very beginning, all weights w_{ij} and biases b_j are assigned small random values using normal or uniform distribution.

2. **Forward Pass:** We feed the input through the network, then each neuron computes its weighted sum and activation. Finally, the network gives some prediction $\hat{y}$.

3. **Computing Loss:** Compare prediction with the true output value y and a loss function is used to measure the error.

4. **Back Propagation:** The network calculates gradients of loss w.r.t. each weight and bias using chain rule. Gradient indicates that how much a small change in weight would increase or decrease the loss.

5. **Gradient Descent:** Update each weight in the direction that reduces the loss:
$$w_{ij}^{new} = w_{ij}^{old} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}}, \qquad b_j^{new} = b_j^{old} - \eta \frac{\partial \mathcal{L}}{\partial b_j}, \tag{3}$$
where η is a small positive number and denotes learning rate.

6. **Iterations:** Steps 2 to 5 are repeated many number of times called as iterations or epochs. With the time, the weights and biases are adjusted so that the network prediction $\hat{y}$ gets closer to output y and hence loss gets minimized.

The architecture of neural network is shown in Figure 1.

2.3 Loss Function Formulation

The total loss was defined as a weighted sum of the physics residual loss and the initial condition loss:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{phy}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}}, \tag{4}$$

with weighting factor $\lambda_{\text{IC}} = 10.0$.

The physics loss was computed from the mean squared residual of the governing equations evaluated at a fixed set of $N = 200$ uniformly spaced collocation points $t_i \in [0, 16]$:

$$\mathcal{L}_{\text{phy}} = \frac{1}{N} \sum_{i=1}^{N} \Big(r_1(t_i)^2 + r_2(t_i)^2 \Big), \tag{5}$$

where the residuals are:

$$\begin{aligned} r_1(t) &= \ddot{x}_1(t) + x_1(t) - 0.5\big(x_2(t) - x_1(t)\big) \\ &+ 0.01\, x_1(t)^3, \end{aligned}$$

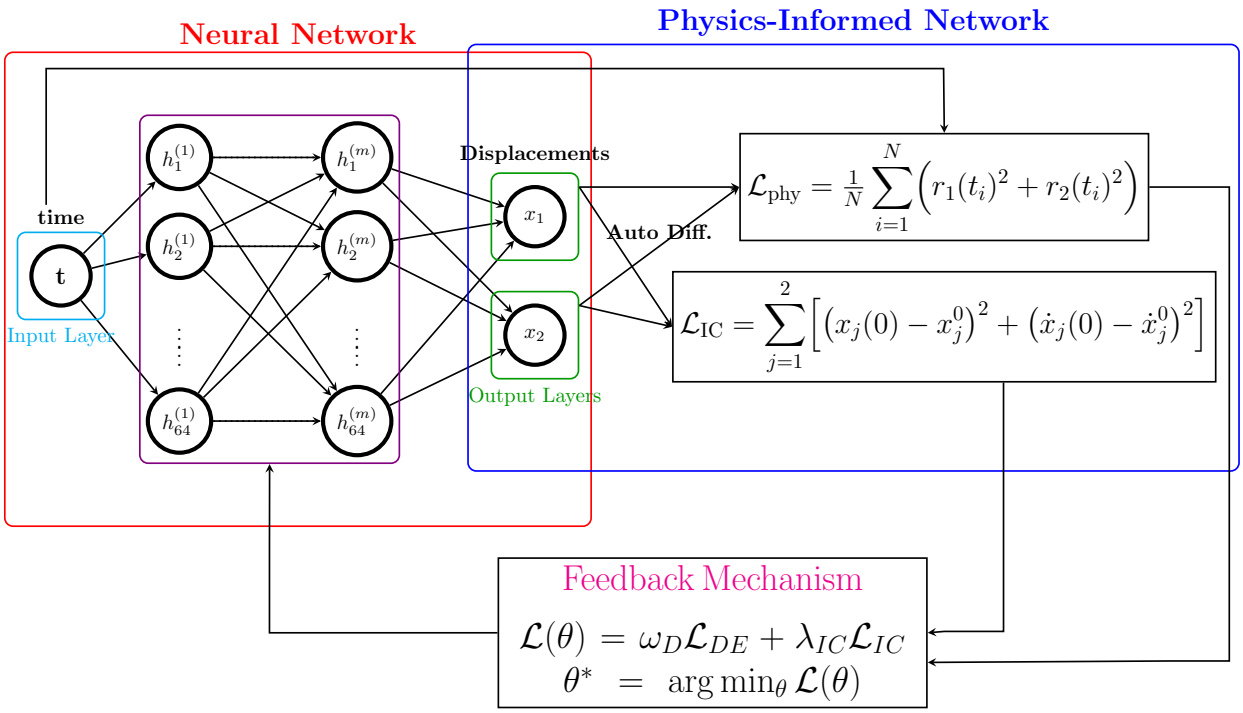


Figure 1: Physics Informed Neural Network Building Blocks

$$\begin{aligned} r_2(t) &= \ddot{x}_2(t) + x_2(t) - 0.5\big(x_1(t) - x_2(t)\big) \\ &+ 0.01\, x_2(t)^3. \end{aligned}$$

First and second-order time derivatives were obtained via automatic differentiation. The initial condition loss enforced the four prescribed values at $t = 0$:

$$\mathcal{L}_{\text{IC}} = \sum_{j=1}^{2}\left[\left(x_j(0) - x_j^0\right)^2 + \left(\dot{x}_j(0) - \dot{x}_j^0\right)^2\right] \tag{6}$$

where $x_1^0 = 1$, $x_2^0 = 0$, $\dot{x}_1^0 = \dot{x}_2^0 = 0$.

Now equation (6) can be expanded as:

$$\begin{aligned} \mathcal{L}_{\text{IC}} &= \big(x_1(0) - 1\big)^2 + \big(x_2(0) - 0\big)^2 \\ &+ \big(\dot{x}_1(0) - 0\big)^2 + \big(\dot{x}_2(0) - 0\big)^2 \end{aligned} \tag{7}$$

2.4 Training Procedure

1. **Optimizer:** Adam optimizer is used with learning rate 10^{-3}.

2. **Collocation points:** 200 uniform points are used within the interval $[0, 16]$

3. **Training steps:** Model is trained over 50,000 iterations.

4. Training history is saved after every 5000 iterations.

5. Intermediate predictions are plotted every 25,000 iterations.

3 Results and Discussion

In this section, we have validated the PINNs solution for coupled spring-mass system using PyTorch. All computations were performed using 64-bit floating-point precision (`torch.float64`) on 11th Gen Intel(R) Core(TM) i5-11300H CPU).

The composite loss function converges steadily over the 50,000 training iterations. Table 1 shows the evolution of the total loss, physics loss, and initial-condition loss on a logarithmic scale. The total loss drops from an initial value of approximately 8–9 to a final value of $\mathbf{1.3771 \times 10^{-5}}$. After 5000 iterations the physics residual becomes the dominant term, while the initial-condition loss remains fluctuating between 10^{-7} and 10^{-5} throughout the later stages.

Epoch	Total Loss	Physics Loss	IC Loss
0	8.5194	0.0813	0.8438
5000	0.0248	0.0247	7.29856×10^{-6}
10000	0.0211	0.0210	5.55193×10^{-6}
15000	0.0169	0.0168	3.41986×10^{-6}
20000	0.0115	0.0113	2.38601×10^{-5}
25000	0.0057	0.0057	9.11852×10^{-7}
30000	0.0002	0.0002	2.14487×10^{-6}
35000	0.0001	0.0001	7.78829×10^{-7}
40000	0.0004	0.0002	2.18778×10^{-5}
45000	0.0007	0.0006	2.00658×10^{-6}
50000	1.3771×10^{-5}	1.23877×10^{-5}	1.3833×10^{-7}

Table 1: Training history of total loss, physics loss, and IC loss.

Figure 2 compares the solution pair $x_1(t)$ and $x_2(t)$ obtained through PINNs predictions represented by solid lines, against the traditional solution obtained through SciPy's `odeint` represented by dashed lines. The two solutions are visually indistinguishable over $t \in [0, 16]$. The nonlinear beating phenomenon is accurately reproduced.

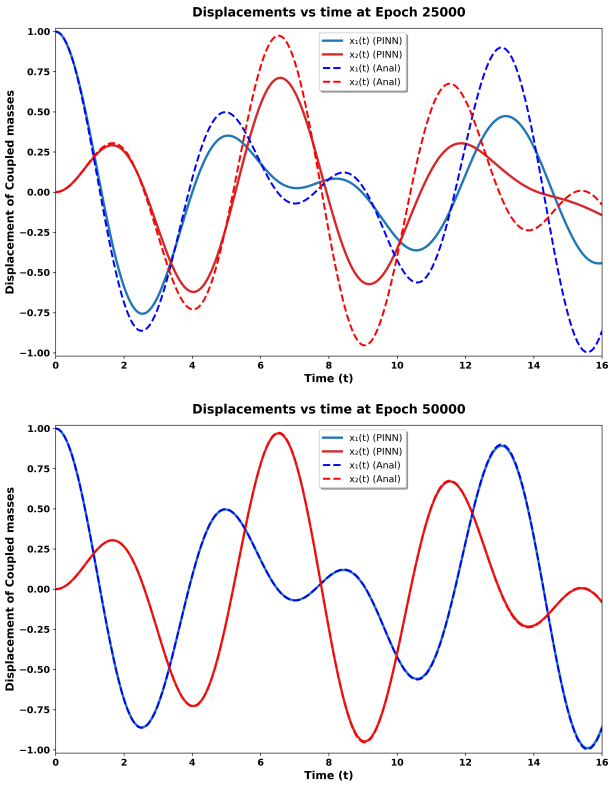


Figure 2: Comparison of PINN-predicted displacements (solid) and numerical reference (dashed) at the final epoch.

The results confirm that a compact PINN can solve the coupled nonlinear oscillator problem to sub-millimeter accuracy using only the governing equations and initial conditions. The network captures both the fast oscillation and the slow energy-transfer timescales without any numerical stiffness.

Advantages of using PINNs are mesh-free formulation, instant evaluation at arbitrary times, and fully differentiable so-

lutions useful for further analysis. For stronger nonlinearities or longer times, adaptive collocation points or larger networks may be required. This study serves as a reproducible benchmark for PINNs performance on nonlinear multiple degree of freedom mechanical systems.

4 Conclusion

This work has demonstrated that PINNs can accurately solve the forward problem of a two-mass coupled nonlinear oscillator governed by second-order ODEs with cubic stiffness nonlinearity. Using a compact fully-connected network and a composite loss function that enforces both the physics residuals and the initial conditions, we obtained solutions that match a high-accuracy numerical solution. The approach is fully mesh-free, differentiable, and instantly evaluable at arbitrary times, making it particularly attractive for nonlinear mechanical systems where traditional numerical integrators may suffer from stiffness or require careful step-size control. The results provide a reproducible benchmark for PINNs performance on multiple degree of freedom nonlinear vibration problems where traditional numerical methods fail to find the solution.

References

[1] J. Xu, J. Ran, and J. She, "The improved pinns for solving partial differential equations with nonlinear couple systems," *Electronic Research Archive*, vol. 34, no. 1, pp. 113–159, 2026.

[2] Raissi *et al. Journal of Computational physics*, vol. 378, 2019.

[3] A. Sharma, A. Awasthi, I. Kant, and O. Sastri, "Solving phase equation using neural network," in *Proceedings of the DAE Symp. on Nucl. Phys*, vol. 69, p. 1219, 2025.

[4] R. Bivins, N. Metropolis, and J. R. Pasta, "Nonlinear coupled oscillators: Modal equation approach," *Journal of Computational Physics*, vol. 12, no. 1, pp. 65–87, 1973.

[5] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, no. 4, pp. 27–31, 2002.

Physics-Informed Neural Networks (PINNs) for the determination of Vibrational Energy Levels of the HCl molecule.

Tanuja Kashyap[1], Arushi Sharma[1], Ishwar Kant[1]

[1]Department of Physics and Astronomical Science,
Central University of Himachal Pradesh, Dharamshala, India
ktanuja584@gmail.com

Submitted on 05-Feb-2026

Abstract

Understanding vibrational spectra is essential in molecular physics, as they reveal the anharmonic behavior of molecular vibrations and the underlying interatomic forces in diatomic molecules. Analysis of vibrational transitions provides valuable information about molecular potentials and bonding characteristics. In this work, a Physics-Informed Neural Network (PINN) approach is employed to determine the vibrational energy levels of the hydrogen chloride (HCl) molecule using the Morse potential. The vibrational dynamics are described by the one-dimensional time-independent Schrödinger equation, with the fundamental physical laws directly incorporated into the neural network training. The PINNs framework simultaneously learns the vibrational wavefunctions, energy eigenvalues, and Morse potential parameters by minimizing a composite loss function that enforces the Schrödinger equation residual, normalization, orthogonality, and physical dissociation constraints. The predicted vibrational energy levels and optimized potential parameters show strong agreement with experimental and theoretical values, demonstrating the accuracy and reliability of the proposed method. These results establish PINNs as a powerful and data-efficient framework for molecular spectroscopy and computational quantum chemistry.

1 Introduction

The study of molecular vibrations plays a central role in understanding molecular structure, bonding, and spectroscopic properties [1, 2, 3, 4].Vibrational energy levels provide detailed information about the potential energy surface and the forces governing atomic motion within molecules. Accurate determination of vibrational spectra is therefore crucial in molecular physics, quan-

tum chemistry, and infrared spectroscopy, particularly for identifying anharmonic effects and dissociation behavior [5, 6, 7, ?, 8].

Conventional numerical techniques, such as matrix diagonalization, finite difference schemes, and Variational Monte Carlo methods, have been widely used to solve the vibrational Schrödinger equation [9, 10, 11, 12]. Although effective, these methods often require carefully chosen basis sets, dense spatial grids, or repeated parameter tuning. Physics-Informed Neural Networks (PINNs) offer a modern alternative by embedding physical laws directly into the learning process, allowing solutions to be obtained in a mesh-free and data-efficient manner [13, 14, 15, 16].

In this work, PINNs are applied to determine the vibrational energy levels of the HCl molecule using the Morse potential [17, 2, 1]. The network simultaneously learns vibrational wave functions, energy eigenvalues, and molecular potential parameters, providing a unified and physically consistent framework for modeling molecular vibrations.

2 Methodology

To determine the vibrational energy levels of the HCl molecule, we employ a Physics-Informed Neural Network (PINNs) framework to solve the one-dimensional time-independent Schrödinger equation governing molecular vibrations. Within the Born–Oppenheimer approximation, the vibrational motion of a diatomic molecule is decoupled from its electronic and rotational degrees of freedom, allowing the system to be described by a single internuclear coordinate [1].

The vibrational dynamics are governed by the time-independent Schrödinger equation:

$$-\frac{\hbar^2}{2\mu}\frac{d^2\psi(r)}{dr^2} + V(r)\psi(r) = E\psi(r), \quad (1)$$

where r denotes the internuclear separation, μ is the reduced mass of the HCl molecule [18], $\psi(r)$ is the vibrational wave function, and E is the corresponding vibrational energy eigenvalue. All calculations are carried out in atomic units by setting $\hbar = 1$.

To account for the anharmonic nature of molecular vibrations and dissociation effects, the interatomic interaction is modeled using the Morse potential [17, 19]:

$$V(r) = D_e\left(e^{-2a(r-r_e)} - 2e^{-a(r-r_e)}\right), \quad (2)$$

where D_e is the dissociation energy, r_e is the equilibrium bond length, and a controls the width of the potential well. In the present framework, the parameters D_e, a, and r_e are treated as trainable variables, enabling the potential energy surface to be learned self-consistently during the training process.

Within the PINNs approach, the vibrational wavefunction is approximated by a fully connected feed-forward neural network:

$$\psi(r) \approx \psi_\theta(r), \tag{3}$$

where θ denotes the network parameters. The trial wavefunction $\psi_\theta(r)$ is represented by a fully connected feed-forward neural network. The internuclear distance r is used as the single input to the network, and the output corresponds to the value of the vibrational wavefunction at that coordinate. The network consists of one input layer, three hidden layers with 64 neurons each, and a single output neuron. Hyperbolic tangent (tanh) activation functions are employed in all hidden layers to ensure smoothness and continuous differentiability of the wavefunction. Spatial derivatives required by the Schrödinger equation are computed using automatic differentiation, ensuring high accuracy and avoiding numerical discretization errors [13, 15]. The output layer uses a linear activation function, allowing the wavefunction to assume both positive and negative values.

The Hamiltonian operator acting on the network-predicted wavefunction is written as

$$\hat{H}\psi(r) = -\frac{1}{2\mu}\frac{d^2\psi(r)}{dr^2} + V(r)\psi(r), \tag{4}$$

and the vibrational energy eigenvalue is obtained directly within the PINNs framework. In this approach, the energy E is introduced as an independent trainable parameter and is learned simultaneously with the vibrational wavefunction during the optimization process.

The neural network parameters and the energy eigenvalue are optimized by minimizing the residual of the Schrödinger equation over the entire spatial domain:

$$\mathcal{L}_{\mathrm{SE}} = \left\langle \left(\hat{H}\psi(r) - E\psi(r)\right)^2 \right\rangle, \tag{5}$$

where $\langle\cdot\rangle$ denotes averaging over collocation points in the internuclear separation domain. As training proceeds, the energy eigenvalue adjusts automatically to minimize the physics-based loss, emerging self-consistently from the enforcement of the Schrödinger equation, boundary conditions, and normalization constraints. No variational principle, matrix diagonalization, or post-processing energy functional is required, making the determination of vibrational energies fully unsupervised and intrinsic to the PINNs formulation [13, 15, 16].

The problem is defined over a finite internuclear domain $r \in [r_{\min}, r_{\max}]$, chosen sufficiently large to ensure that the vibrational wavefunction decays to zero at the boundaries. The normalization condition for bound vibrational states,

$$\int_{r_{\min}}^{r_{\max}} \psi^2(r), dr = 1, \tag{6}$$

is enforced explicitly during training. Higher vibrational states are obtained sequentially by imposing orthogonality constraints between the current wavefunction and all previously computed lower-energy states.

The total PINN loss function is a composite of the physics-informed residual, nor-

malization, and orthogonality constraints [13, 15, 16]:

$$\mathcal{L} = \mathcal{L}_{\mathrm{SE}} + \mathcal{L}_{\mathrm{N}} + \mathcal{L}_{\mathrm{O}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{SE}}$ penalizes the residual of the Schrödinger equation, $\mathcal{L}_{\mathrm{N}}$ enforces wavefunction normalization, and $\mathcal{L}_{\mathrm{O}}$ ensures orthogonality between different vibrational states. Vibrational eigenstates are obtained sequentially, starting from the ground state, while the Morse potential parameters are optimized to reproduce the observed vibrational energy spacings.

This physics-informed formulation allows the vibrational spectrum and molecular potential of HCl to be determined in a unified and fully unsupervised manner, without requiring predefined basis functions or explicit matrix diagonalization. The smooth and noise-free evaluation of wavefunctions and their derivatives through automatic differentiation ensures a continuous representation of the solution over the entire domain.

3 Results and Discussion

The Physics-Informed Neural Network (PINNs) successfully solves the vibrational Schrödinger equation for the HCl molecule and converges to physically meaningful eigenstates. The predicted vibrational energy levels show excellent agreement with experimental data. The decreasing spacing between successive energy levels correctly reflects the anharmonic nature of molecular vibrations captured by the Morse potential.

The vibrational eigenstates are obtained sequentially, starting from the ground state to v=3. For each vibrational level, the neural network learns the corresponding wavefunction while the Morse potential parameters are optimized to reproduce the observed vibrational energy spacings. Orthogonality constraints are imposed between the current wavefunction and all previously computed lower-energy states [13, 15, 16].

The network is trained by minimizing a composite loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{SE}} + \mathcal{L}_{\mathrm{N}} + \mathcal{L}_{\mathrm{O}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{SE}}$ penalizes the residual of the Schrödinger equation, $\mathcal{L}_{\mathrm{N}}$ enforces wavefunction normalization, and $\mathcal{L}_{\mathrm{O}}$ ensures orthogonality between different vibrational states. The Schrödinger residual loss is defined as

$$\mathcal{L}_{\mathrm{SE}} = \left\langle \left(\hat{H}\psi(r) - E\psi(r)\right)^2 \right\rangle, \tag{9}$$

where $\langle\cdot\rangle$ denotes averaging over the training domain. This physics-informed formulation allows the vibrational spectrum and molecular potential of HCl to be determined in a unified and fully unsupervised manner, without requiring predefined basis functions or explicit matrix diagonalization.

A key advantage of the PINNs approach is the smooth and noise-free evaluation of wavefunctions and their derivatives through automatic differentiation [13, 15, 16]. Unlike finite-difference or Numerov

methods [20], this framework avoids grid-induced artifacts and provides a continuous representation of the solution over the entire domain. The optimized Morse potential parameters are consistent with known spectroscopic values and exhibit the correct asymptotic dissociation behavior [2, 18].

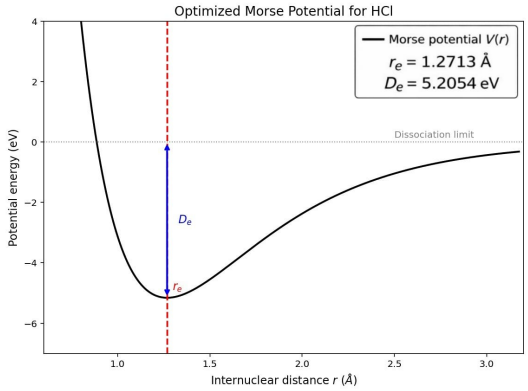


Figure 1: Optimized Morse potential $V(r)$ for HCl obtained from the PINNs. The potential minimum corresponds to the equilibrium bond length r_e, and the dissociation energy D_e is indicated.

The optimized Morse potential parameters obtained from the PINNs are: $D_e = 5.2054$ eV, $a = 1.7551$ Å^{-1}, and $r_e = 1.2713$ Å. As depicted in Figure1, the potential curve obtained from the PINNs closely matches the theoretical shape of a Morse potential, validating the accuracy of the predicted parameters

The vibrational energy levels predicted by the PINNs are summarized in Table 1 . Energies are expressed in wavenumbers (cm^{-1}) using the relation $\tilde{\nu}_v = E_v/(hc)$.

These results show the expected anharmonic spacing between successive vibrational levels, decreasing as v increases, which is characteristic of the Morse potential. The PINNs framework thus provides both the potential energy curve and the vibrational spectrum in a fully unsupervised and physically consistent manner.

Table 1: HCl vibrational frequencies: experimental vs PINNs-predicted (cm^{-1}).

Transition	Exp.(cm^{-1})	Predicted (cm^{-1})
$0 \rightarrow 1$	2885.9764	2892.2881
$0 \rightarrow 2$	5667.983	5615.8731
$0 \rightarrow 3$	8356.7769	8635.5715

4 Conclusion

This study demonstrates the effectiveness of Physics-Informed Neural Networks for determining vibrational energy levels of diatomic molecules [13, 15, 16]. By embedding the time independent Schrödinger equation and physical constraints directly into the learning process, the PINNs accurately predicts vibrational wavefunctions, energy eigenvalues, and Morse potential parameters for the HCl molecule [17, 2, 18].

The network successfully reproduces the ground-state and higher vibrational states [1, 3]. The predicted energy levels are in excellent agreement with experimental data, capturing the decreasing spacing between successive levels due to the anharmonicity of molecular vibrations [5, 6].

The Morse potential is learned self-consistently during training, yielding parameters consistent with known spectro-

scopic values and correctly describing the dissociation behavior of HCl [2, 17]. The PINNs approach provides smooth, noise-free wavefunctions and derivatives through automatic differentiation, avoiding the limitations of traditional grid-based or matrix diagonalization methods [13, 15, 16].

Overall, this work highlights the PINNs framework as a flexible, data efficient, and fully unsupervised method for molecular spectroscopy and computational quantum chemistry [13, 15]. This methodology can be extended to more complex molecular systems and higher-dimensional quantum problems, providing a powerful tool for modeling vibrational dynamics and potential energy surfaces in diatomic and polyatomic molecules.

References

[1] G. Herzberg. *Molecular Spectra and Molecular Structure: I. Spectra of Diatomic Molecules*. Van Nostrand, New York, 1950.

[2] K. P. Huber and G. Herzberg. *Constants of Diatomic Molecules*. Van Nostrand Reinhold, New York, 1979.

[3] C. N. Banwell and E. M. McCash. *Fundamentals of Molecular Spectroscopy*. McGraw-Hill, 1994.

[4] W. S. Struve. *Fundamentals of Molecular Spectroscopy*. Wiley, New York, 1989.

[5] A. Sharma and O. S. K. S. Sastri. Numerical simulation of ro-vibrational spectra for diatomic molecules using the numerov matrix method. *European Journal of Physics*, 43(1):015404, 2022.

[6] A. Sharma, O. S. K. S. Sastri, S. Awasthi, A. Khachi, and L. Kumar. Simulation of vibrational spectrum of diatomic molecules using morse potential by matrix methods in gnumeric worksheet. *Physics Education*, 2022.

[7] A. Sharma and O. S. K. S. Sastri. Numerical solution of schrödinger equation for rotating morse potential using matrix methods with fourier sine basis. *International Journal of Quantum Chemistry*, 121:e26682, 2021.

[8] A. Sharma and O. S. K. S. Sastri. Numerical simulation of quantum anharmonic oscillator using matrix methods. *European Journal of Physics*, 2020.

[9] F. Marsiglio. The harmonic oscillator in quantum mechanics: A third way. *American Journal of Physics*, 77:253–258, 2009.

[10] O. S. K. S. Sastri et al. Numerical solution of square well potential with matrix method using worksheets. *Physics Education*, 2019.

[11] O. Bayrak and I. Boztosun. Arbitrary ℓ-state solutions of the rotating morse potential by the asymptotic iteration method. *Journal of Physics A: Mathematical and General*, 39:6955–6963, 2006.

[12] V. Sharda, O. S. K. S. Sastri, J. Bhardwaj, and A. K. Jha. A computer simulation using spreadsheets for learning concept of steady-state equilibrium. *Physics Education*, 51(2):025007, 2016.

[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*, 378:686–707, 2019.

[14] M. Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19:1–24, 2018.

[15] Z. Chen, Y. Liu, and S. Sun. Physics-informed neural networks for solving quantum schrödinger equations. *Computer Methods in Applied Mechanics and Engineering*, 376:113581, 2021.

[16] S. Mishra and R. Molinaro. Estimates on the generalization error of physics-informed neural networks. *SIAM Journal on Numerical Analysis*, 59:1751–1779, 2021.

[17] P. M. Morse. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Physical Review*, 34:57–64, 1929.

[18] National Institute of Standards and Technology. Nist atomic spectroscopy compendium. `https://www.nist.gov`.

[19] C. Berkdemir and J. Han. Rotational correction on the morse potential through pekeris approximation. *arXiv preprint arXiv:quant-ph/0502182*, 2005.

[20] LibreTexts Chemistry. Harmonic oscillators and ir spectroscopy. `https://chem.libretexts.org`.

Physics Informed Neural Network Approach for Estimating the Coefficients of the Semi-Empirical Mass Formula

Anjali[1] and Ayushi Awasthi[1]

[1]Department of Physics and Astronomical Science,
Central University of Himachal Pradesh (CUHP),
Dharamshala, Himachal Pradesh 176215, India.

Submitted on 08-02-2026

Abstract

Background: The Semi-Empirical Mass Formula (SEMF), also known as the Weizsäcker formula, provides an approximate description of nuclear binding energy by incorporating volume, surface, Coulomb, asymmetry, and pairing effects within the liquid-drop model framework. Conventionally, the coefficients of the SEMF are determined using least-squares regression on experimental nuclear mass data.

Purpose: The purpose of this study is to develop and demonstrate a Physics-Informed Neural Networks (PINNs) framework for estimating the coefficients of the SEMF by embedding the nuclear-physics structure of the model directly into the learning process.

Methods:A physics-informed neural network was developed with the SEMF explicitly embedded in the loss function. Experimental nuclear mass data were used for training, while the SEMF coefficients were treated as trainable parameters. The model was optimized via c^2 minimization, measuring weighted squared deviations between experimental and predicted binding energies, with physically motivated bounds imposed to ensure stability and interpretability.

Results: The PINN framework successfully recovered physically meaningful values of the SEMF coefficients with improved numerical stability compared to conventional regression-based methods. The predicted binding energies exhibited strong agreement with experimental data across a wide range of nuclei. Incorporation of physical constraints significantly reduced overfitting and enhanced the generalization capability of the model.

Conclusion: This study demonstrates that Physics-Informed Neural Networks provide an effective and reliable framework for estimating the coefficients of the Semi-Empirical Mass Formula.

1 Introduction

The atomic nucleus is a complex many-body system governed by the strong nuclear force, and its properties cannot be derived exactly from first principles for most nuclei. To address this challenge, several phenomenological models have been developed to describe nuclear structure and binding energies. One of the earliest and most influential approaches is the Liquid Drop Model (LDM), originally proposed by George Gamow[1] in 1929, in which the nucleus is treated as an incompressible drop of nuclear fluid composed of protons and neutrons. This model successfully captures the collective behavior of nucleons and explains macroscopic nuclear properties such as binding energy, stability, and fission. Gamow also provided a quantum-mechanical explanation of alpha decay, further strengthening the physical foundation of the liquid drop analogy. Building upon the liquid drop picture, the Semi-Empirical Mass Formula (SEMF), also known as the Weizsäcker formula,[2] provides an approximate expression for nuclear binding energy by incorporating five key contributions: volume, surface, Coulomb, asymmetry, and pairing energies. These terms account for both theoretical considerations and empirical observations, allowing the SEMF to reproduce general trends in nuclear masses across the chart of nuclides. The formula represents a mathematical realization of the liquid drop concept, where the total binding energy emerges from the collective contribution of these energy terms, each weighted by a corresponding coefficient.

Traditionally, the coefficients of the SEMF and the Liquid Drop Model are determined by fitting experimental nuclear mass data using least-squares minimization (LSM)[3] techniques. While regression-based approaches are computationally efficient and widely used, they suffer from certain limitations. In least-squares fitting, data points with large deviations can disproportionately influence the optimized parameters, potentially biasing the resulting coefficients. Furthermore, such methods rely heavily on the quality and distribution of the available mass data.

In recent years, stochastic optimization techniques such as Variational Monte Carlo (VMC)[3] have been investigated as alternative methods for determining the coefficients of the Semi-Empirical Mass Formula (SEMF). While VMC provides greater flexibility and avoids strict analytical assumptions, its performance strongly depends on the choice of initial parameter values. The optimization process is sensitive to the initial guess of the coefficients, which can lead to convergence toward different local minima and result in variability across independent runs. These limitations highlight the need for a more robust and stable optimization framework that can reduce dependence on initial conditions and improve reproducibility.

In this work, we determine the coefficients of the Semi-Empirical Mass For-

mula using Physics-Informed Neural Networks (PINNs), treating the PINN framework as an optimization technique. By embedding the underlying physical structure of the SEMF directly into the neural network loss function, the PINN approach enables efficient and stable parameter estimation while maintaining consistency with nuclear physics constraints[4].

2 Methodology

From the LDM,the theoretical binding energy of a nucleus with mass number A_i and proton number Z_i is described using the Semi-Empirical Mass Formula SEMF[3]. The binding energy is expressed as the sum of different energy contributions,

$$B_i^{\text{th}}(A_i, Z_i) = a_v A_i - a_s A_i^{2/3} - a_c \frac{Z_i(Z_i-1)}{A_i^{1/3}} - a_a \frac{(A_i - 2Z_i)^2}{A_i} + \delta_i \, a_p A_i^{-1/2}, \tag{1}$$

where ,

- a_v : volume energy coefficient,
- a_s : surface energy coefficient,
- a_c : Coulomb energy coefficient,
- a_a : asymmetry energy coefficient,
- a_p : pairing energy coefficient.

The pairing factor δ_i accounts for odd–even effects and is defined according to the parity of the proton and neutron numbers[3].

Within the Physics-Informed Neural Network (PINNs) framework, the SEMF coefficients are treated as trainable parameters and collected into a single parameter vector[5],

$$\boldsymbol{a} = \begin{bmatrix} a_v & a_s & a_c & a_a & a_p \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

Here T in superscript is an trainable coefficient vector,each component of $\boldsymbol{a}$ retains a clear physical interpretation and corresponds to a specific contribution to the nuclear binding energy.

2.1 PINN Architecture for SEMF Parameter Estimation

Unlike conventional applications of PINNs, which are typically used to solve partial differential equations [6], in this work we employ the PINNs framework for parameter estimation of the SEMF. The framework for parameter estimation is illustrated in Fig. 1. The network input consists of experimentally measured nuclear properties[7],namely the mass number A_i and proton number Z_i, which are taken from the AME 2020 atomic mass evaluation [8]. The network does not learn the binding energy directly. Instead, the SEMF coefficients $\boldsymbol{a}$ are optimized during training.

During the forward pass, the analytical SEMF expression is evaluated using the current set of coefficients to compute the predicted binding energy B_{i}^{SEMF}. This approach ensures that the physical structure of the SEMF is explicitly preserved throughout the learning process.

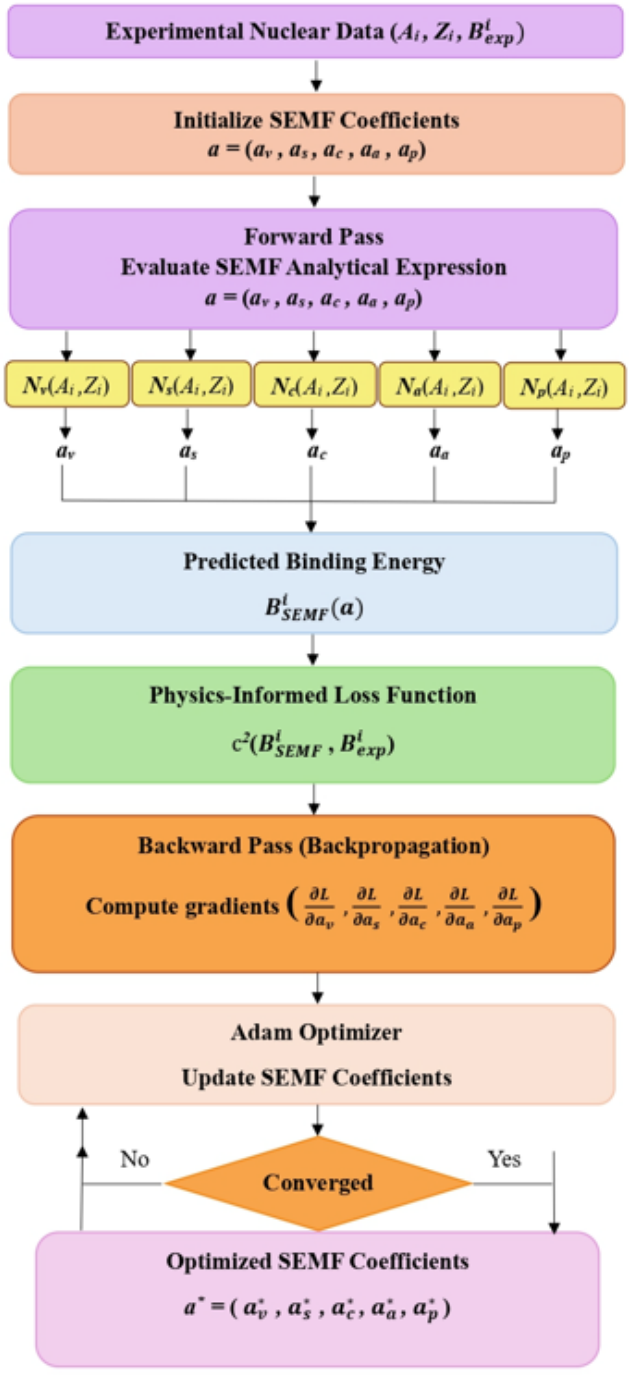


Figure 1: Flowchart for obtaining coefficints of SEMF using PINNs

In the Physics-Informed Neural Network (PINNs) framework adopted in this work.The estimation of the Semi-Empirical Mass Formula (SEMF) coefficients is carried out using a parallel-network architecture. Instead of employing a single neural network to infer all coefficients simultaneously, independent neural networks are assigned to each SEMF coefficient. This design choice preserves the physical interpretability of the parameters and avoids strong coupling between unrelated energy terms.

The SEMF coefficient vector is defined as

$$\boldsymbol{a} = \begin{bmatrix} a_v & a_s & a_c & a_a & a_p \end{bmatrix}^{\mathrm{T}}, \tag{3}$$

where each coefficient corresponds to a distinct physical contribution to the nuclear binding energy.

Within the PINN architecture, five parallel feedforward neural networks are employed:

$$a_v = \mathcal{N}_v(A_i, Z_i), \tag{4}$$
$$a_s = \mathcal{N}_s(A_i, Z_i), \tag{5}$$
$$a_c = \mathcal{N}_c(A_i, Z_i), \tag{6}$$
$$a_a = \mathcal{N}_a(A_i, Z_i), \tag{7}$$
$$a_p = \mathcal{N}_p(A_i, Z_i), \tag{8}$$

where $\mathcal{N}_v$, $\mathcal{N}_s$, $\mathcal{N}_c$, $\mathcal{N}_a$, and $\mathcal{N}_p$ denote five independent neural networks with identical architectures but separate trainable parameters.

Each network receives the same physical inputs, namely the mass number A_i and proton number Z_i, and outputs a single scalar corresponding to one SEMF co-

efficient. The outputs of these parallel networks are then embedded directly into the analytical SEMF expression to compute the predicted binding energy:

$$B_i^{\mathrm{SEMF}} = \mathcal{N}_v A_i - \mathcal{N}_s A_i^{2/3} - \mathcal{N}_c \frac{Z_i(Z_i-1)}{A_i^{1/3}} - \mathcal{N}_a \frac{(A_i-2Z_i)^2}{A_i} + \delta_i \mathcal{N}_p A_i^{-1/2}. \tag{9}$$

The parallel-network structure ensures that each SEMF coefficient is optimized independently while remaining constrained by the governing physical model. During training, a single physics-informed loss function is minimized, and gradient-based optimization[9] updates the parameters of all five networks simultaneously.

2.2 Working Principle of the Physics-Informed Neural Network

The PINNs operates through a physics-guided optimization strategy. For each nucleus characterized by (A_i, Z_i), the SEMF-based binding energy is computed using the current parameter vector $\boldsymbol{a}$. The difference between the predicted and experimentally measured binding energies is then quantified using a physics-informed loss function.

Unlike conventional neural networks that employ hidden layers to learn implicit representations, the present architecture performs optimization directly on physically meaningful parameters. Gradient-based optimization algorithms such as adam optimizer[10] are employed to iteratively update the SEMF coefficients until convergence is achieved.

This formulation ensures that the learned parameters remain consistent with nuclear-physics principles and avoids the emergence of unphysical solutions. The loss function is constructed by comparing the SEMF-based predictions with experimental binding energy data. A c^2 - square loss is defined as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left(B_i^{\mathrm{SEMF}} - B_i^{\mathrm{exp}}\right)^2}{B_i^{\mathrm{exp}}}, \tag{10}$$

where N denotes the total number of nuclei in the training data set, and B_i^{exp} represents the experimentally measured binding energy.

In analogy with the residual loss used in PDE-based PINNs, this loss function enforces consistency between the physical model and experimental observations. The present framework therefore addresses an inverse problem in which the physics-informed constraint is algebraic rather than differential.

Within the PINNs framework, the optimization of the Semi-Empirical Mass Formula (SEMF) coefficients is carried out using backpropagation[11]. Unlike conventional neural networks, backpropagation does not propagate errors through an arbitrary mapping between inputs and outputs. Instead, the gradients are propagated through the analytical SEMF expression itself.

During training, the gradients of the loss function with respect to each SEMF coefficient are computed using backpropaga-

tion,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}} = \left(\frac{\partial \mathcal{L}}{\partial a_v}, \frac{\partial \mathcal{L}}{\partial a_s}, \frac{\partial \mathcal{L}}{\partial a_c}, \frac{\partial \mathcal{L}}{\partial a_a}, \frac{\partial \mathcal{L}}{\partial a_p} \right). \quad (11)$$

These gradients propagate through the SEMF analytical structure, ensuring that updates to the coefficients are consistent with the underlying nuclear-physics model. Gradient-based optimization algorithms, such as stochastic gradient descent or adaptive optimizers, use these gradients to update the parameters iteratively[5],

$$\boldsymbol{a}^{(k+1)} = \boldsymbol{a}^{(k)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}}, \quad (12)$$

where η denotes the learning rate and k represents the iteration index.

By embedding backpropagation directly within the SEMF formulation, the PINN ensures that learning is guided simultaneously by experimental data and established nuclear-physics constraints. As a result, the optimized coefficients remain physically interpretable while achieving optimal agreement with experimental binding energies.

3 Results and Discussion

The Physics-Informed Neural Network (PINNs) framework successfully converged to a stable and well-defined set of coefficients for the Semi-Empirical Mass Formula (SEMF). The optimized coefficients were found to be physically reasonable and consistent with the expected orders of magnitude reported in the literature. Moreover, the robustness of these coefficients across multiple independent training runs demonstrates that the incorporation of physical constraints within the loss function[9] effectively regularizes the learning process and mitigates overfitting.

To quantitatively access the accuracy of the obtained coefficients, the deviation between theoretical and experimental nuclear binding energies was evaluated using the c-square minimization criteria. This metric penalizes large discrepancies between predicted and observed values while preserving the original physical units (MeV), thereby providing a meaningful measure of model performance.

The c^2-square error is defined as

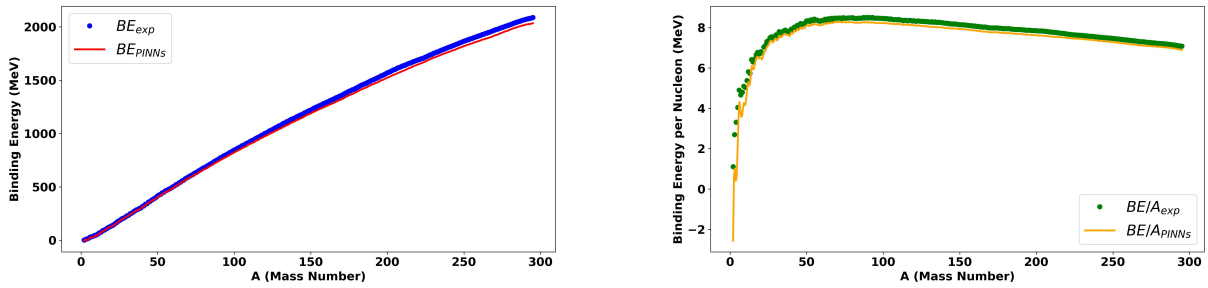
$$c^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(B_i^T - B_i^E\right)^2}{B_i^E}, \quad (13)$$

where B_i^T and B_i^E represent the theoretical and experimental binding energies of the i-th nucleus, respectively, and n denotes the total number of nuclei included in the dataset.

The optimized SEMF coefficients are recovered after embedding the physical constraints into the PINN architecture are given by

$$\begin{aligned} a_v &= 15.072 MeV, \\ a_s &= 16.597 MeV, \\ a_c &= 0.690 MeV, \\ a_a &= 20.907 MeV, \\ a_p &= 12.605 MeV. \end{aligned} \quad (14)$$

The comparison between experimental nuclear binding energies and those pre-



(a) (b)

Figure 2: (a) Comparison of experimental nuclear binding energies with those predicted by the Semi-Empirical Mass Formula using PINN-optimized coefficients. (b) Comparison of experimental nuclear binding energies per nucleon with those predicted by the Semi-Empirical Mass Formula using PINN-optimized coefficients, both as a function of mass number A.

dicted by the Semi-Empirical Mass Formula (SEMF) using PINN-optimized coefficients is shown in Fig. 2b. Similarly, Fig. 2b presents the comparison of experimental binding energy per nucleon with the values predicted by the SEMF, both plotted as a function of mass number A. The PINNs predictions closely follow the experimental trends across a wide range of mass numbers A, indicating that this framework effectively captures the dominant physical contributions governing nuclear binding.

Since the SEMF is explicitly embedded within the model architecture, the learned parameters retain clear physical interpretations, in contrast to purely data-driven neural network approaches where model parameters often lack physical meaning. By combining experimental data with established nuclear-physics principles, the PINNs approach yields improved numerical stability, enhanced predictive capability, and greater physical consistency, highlighting its effectiveness for inverse problems in nuclear physics[9].

4 Conclusion

In this work, the PINNs methodology was applied to the Semi-Empirical Mass Formula to estimate its phenomenological coefficients while preserving the analytical structure and physical foundations of the model. By treating the SEMF coefficients as trainable parameters constrained by the functional form of the mass formula, the PINNs approach ensures that the learned parameters remain physically interpretable and consistent with nuclear theory.During the forward pass, the analytical SEMF expression was evaluated to compute the predicted binding energy, ensuring that the physical form of the model was maintained throughout the learning process. The

backward pass propagated loss gradients through the SEMF formulation itself, allowing gradient-based optimization to update the coefficients in a manner consistent with nuclear-physics constraints[5].The use of parallel neural networks for individual SEMF coefficients enabled independent optimization of each energy term while retaining clear physical interpret-ability. The physics-informed loss function, based on chi-square minimization, ensured consistency between theoretical predictions and experimental observations.The advantage of using PINNS to obatin the SEMF coefficients is that the physical laws of SEMF embedded directly into the optimization process resulting in stable parameter estimates with reduced senstivity to initial conditions compared to purely regression based methods(LSM)[3].

Acknowledgments

A. Awasthi acknowledges the financial support provided by the DST, Govt. of India, through Grant No. DST/INSPIRE Fellowship/2020/IF200538.

References

[1] G. Gamow, "Zur Quantentheorie des Atomkernes," *Zeitschrift für Physik*, vol. 51, pp. 204–212, 1928.

[2] C. F. von Weizsäcker, "Zur Theorie der Kernmassen," *Zeitschrift für Physik*, vol. 96, pp. 431–458, 1935.

[3] Gora, S., Sastri, O. S. K. S., & Soni, S. K., Optimization of semi-empirical mass formula coefficients using least square minimization and variational Monte-Carlo approaches, *European Journal of Physics* **43**(3), 035802 (2022).

[4] S. Cuomo, V. Di Cola, F. Giampaolo, and G. Rozza, "Scientific Machine Learning through Physics-Informed Neural Networks: Where We Are and What's Next," *Journal of Scientific Computing*, vol. 92, pp. 1–35, 2022.

[5] A. Awasthi, S. Walia, S. Gora and O. S. K. S. Sastri, "Optimization of Coefficients of Semi-Empirical Mass Formula Using Physics Informed Neural Networks,"in *Proceedings of the DAE Symposium on Nuclear Physics*, Vol. 69, Paper G14, 2025. :contentReference[oaicite:1]index=1

[6] P. Möller, W. D. Myers, H. Sagawa, and S. Yoshida, "Nuclear Masses and Deformations," *Atomic Data and Nuclear Data Tables*, vols. 109–110, pp. 1–204, 2016.

[7] Gjorgievska, S., Kochankovski, H., Stankovic, K., & Barandovski, L., Revision of the semi-empirical mass formula coefficients using the AME2020 database, *Nuclear Engineering and Design* **426**, 113403 (2024).

[8] Wang, M., Huang, W. J., Kondev, F. G., Audi, G., & Naimi, S., The AME

2020 atomic mass evaluation (II). Tables, graphs and references, *Chinese Physics C*, **45(3)**, 030003 (2021).

[9] 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.

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

Shell Structure of ^{40}Ca from the Woods-Saxon Potential by Solving the Time Independent Schrödinger Equation Using PINNs

Tamanna Saklani[1] and Ayushi Awasthi[1]

[1]Department of Physics and Astronomical Sciences,
Central university of Himachal Pradesh 176215, India

Submitted on 08-Feb-2026

Abstract

Background: The nuclear shell model provides a fundamental framework for understanding nuclear structure and magic numbers by describing nucleons moving in an average mean-field potential. Conventional grid-based solutions of the radial time-independent Schrödinger equation often require careful discretization and parameter tuning. Physics-Informed Neural Networks (PINNs) offer an alternative by embedding physical laws directly into the learning process.

Purpose: The aim of this work is to compute single-particle neutron and proton energy levels of the doubly magic nucleus ^{40}Ca by solving the radial time-independent Schrödinger equation using PINNs and to compare the results with experimental data.

Methods: The effective potential includes the Woods-Saxon central term, centrifugal barrier, spin-orbit interaction, and Coulomb potential for protons. Separate PINNs are trained for the $1s$, $1p$, and $1d$ orbitals, with the energy eigenvalue treated as a trainable parameter. The loss function incorporates the Schrödinger equation residual, wavefunction normalization, and weak experimental guidance, and the network is optimized using the Adam optimizer.

Results: The PINNs framework accurately reproduces bound-state energy levels of neutrons and protons in ^{40}Ca. Neutron energies for the $1s$, $1p$, and $1d$ shells show close agreement with experimental values, with deviations below 0.3 MeV. The model

correctly captures spin-orbit splitting, with $j = \ell + 1/2$ states more deeply bound than their $j = \ell - 1/2$ counterparts. Proton energies exhibit systematic upward shifts due to Coulomb repulsion, with deviations of about 0.5–1.3 MeV for higher orbitals.

Conclusions: These results demonstrate that PINNs provide a robust and mesh-free framework for solving nuclear shell-model problems, offering a reliable alternative to traditional numerical methods for nuclear structure calculations.

1 Introduction

The nuclear shell model provides a fundamental framework for understanding nuclear structure and the emergence of magic numbers by describing nucleons as independent particles moving within an average mean-field potential.[3]For a doubly magic nucleus such as ^{40}Ca, this structure is characterized by the completion of specific energy shells, which necessitates precise calculation of single-particle energy levels.[1] Traditionally, determining these levels requires solving the radial time-independent Schrödinger equation using grid-based numerical methods (like CCD , Matrix Method)that often demand intensive parameter tuning and complex mesh construction to satisfy various physical constraints. However, the advent of Physics-Informed Neural Networks (PINNs) offers a transformative alternative by embedding these physical laws directly into the machine learning architecture. By utilizing a PINNs framework, the Schrödinger residual is integrated into a multi-component loss function alongside wavefunction normalization and orthogonality constraints. For this we have taken the effective potential which consist of Woods-Saxon potential,centrifugal, spin-orbit, and Coulomb terms—to accurately predict neutron and proton energy levels for the $1s$, $1p$, and $1d$ states.In conclusion, the PINNs-based framework provides a flexible approach for nuclear structure calculations. By resolving the radial Schrödinger equation through a physics-informed loss function, the model achieves close agreement with experimental benchmarks while operating independently of traditional grid-dependent numerical solvers.

2 Methodology

The computational framework developed in this study utilizes a Physics-Informed Neural Network (PINNs) to determine the

single-particle energy eigenvalues and corresponding eigenfunctions of the ^{40}Ca nucleus. Unlike traditional grid-based numerical methods, this approach embeds the governing physical laws directly into the neural network's optimization process through the loss function.

The core objective is to solve the radial Time-Independent Schrödinger Equation (TISE) for a nucleon within a mean-field potential:

$$-\frac{\hbar^2}{2\mu}\frac{d^2u}{dr^2} + [V_{\text{eff}}(r) - E]\,u(r) = 0 \qquad (1)$$

where $u(r)$ represents the radial wavefunction, E is the energy eigenvalue to be determined, and μ is the reduced mass of the nucleon.

The effective potential $V_{\text{eff}}(r)$ is constructed as a superposition of four distinct physical components:

- **Woods-Saxon Central Potential ($V_{\mathbf{WS}}$):** Models the short-range nuclear force by defining a mean-field potential with a constant interior depth V_0 that smoothly decays to zero at the nuclear surface R according to the diffuseness parameter a.[5]

$$V_{WS}(r) = \frac{V_0}{1 + \exp\left(\frac{r-R}{a}\right)} \qquad (2)$$

- **Spin-Orbit Coupling (V_{SO}):** Accounts for the $\mathbf{L}\cdot\mathbf{S}$ force that causes the splitting of (n,ℓ) shells into $j = \ell \pm 1/2$ sub-shells, a key mechanism required to reproduce the experimentally observed magic numbers.[5]

$$V_{SO}(r) = V_{so}\left(\frac{\hbar}{m_\pi c}\right)^2 \frac{1}{r}\frac{d}{dr}\left[\frac{1}{1+\exp\left(\frac{r-R}{a}\right)}\right](\mathbf{L}\cdot\mathbf{S}) \qquad (3)$$

- **Centrifugal Barrier ($V_{\mathbf{CF}}$):** Represents the l-dependent repulsive term in the radial Schrödinger equation that accounts for the kinetic energy of orbital motion and regulates the nucleon's proximity to the nuclear core.[5]

$$V_{CF}(r) = \frac{\hbar^2}{2\mu}\frac{l(l+1)}{r^2} \qquad (4)$$

- **Coulomb Potential ($V_{\mathbf{C}}$):** Accounts for the long-range electrostatic repulsion experienced by protons, modeled by treating the nucleus as a sphere of uniform charge density.[5]

$$V_C(r) = \begin{cases} \frac{(Z-1)e^2}{r}, & r \geq R \\ \frac{(Z-1)e^2}{2R}\left(3 - \frac{r^2}{R^2}\right), & r < R \end{cases} \qquad (5)$$

PINNs Architecture

The radial wavefunction $u(r)$ is approximated by a deep feed-forward neural network consisting of two hidden layers with

64 neurons each. We employ the *Tanh* activation function to ensure the higher-order differentiability required for computing second-order derivatives. Uniquely, the energy E is treated as a trainable parameter (`nn.Parameter`),[2] allowing the network to converge toward the physical eigenvalue simultaneously with the wavefunction. The overall workflow of the proposed framework used to solve the radial Schrödinger equation is illustrated in Fig. 1.

The network is trained by minimizing a composite loss function $\mathcal{L}_{\text{total}}$, which enforces adherence to the governing physical equations and relevant constraints,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{physics}} + \lambda_1 \mathcal{L}_{\text{exp}} + \lambda_2 \mathcal{L}_{\text{norm}}. \quad (6)$$

The individual components of the loss function are defined as follows:

1. **Physics Loss ($\mathcal{L}_{\text{physics}}$):** The mean squared residual of the radial time-independent Schrödinger equation evaluated at collocation points within the domain. First- and second-order spatial derivatives of the wavefunction are computed using automatic differentiation, ensuring consistency with the neural network representation [4].

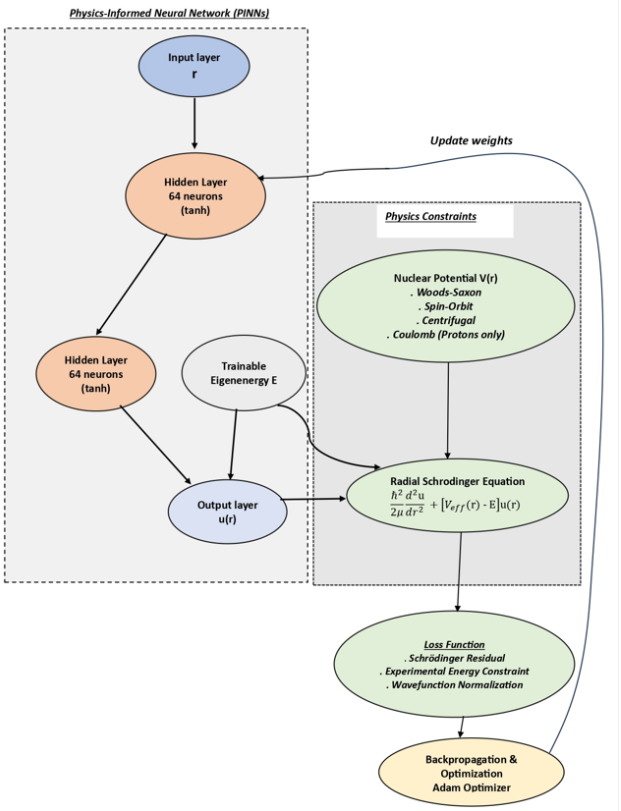


Figure 1: Schematic flowchart of the Physics-Informed Neural Network (PINNs) architecture for solving the radial Schrödinger equation. The model combines a feed-forward neural network with embedded physical constraints through the effective nuclear potential, enabling simultaneous learning of the wavefunction u(r) and the eigen energy E.

2. **Experimental Guidance Loss ($\mathcal{L}_{\text{exp}}$):** A soft constraint that biases the trainable energy parameter E to-

ward known experimental values, thereby improving convergence behavior while preserving the physics-driven nature of the solution.

3. **Normalization Loss ($\mathcal{L}_{\text{norm}}$):** Enforces the normalization condition of the radial wavefunction, $\int |u(r)|^2 \, dr = 1$, preventing convergence to trivial solutions and ensuring a physically meaningful probability interpretation [4].

The weighting coefficients λ_1 and λ_2 regulate the relative contributions of the auxiliary loss terms during optimization and are selected to balance physical fidelity and numerical stability. These coefficients act as hyperparameters of the training process and do not alter the underlying physical model. The network parameters, including weights, biases, and the energy eigenvalue E, are optimized using the Adam (Adaptive Moment Estimation) optimizer. Training is performed over the radial domain $r \in [0.01, 12.0]$ fm for approximately 6000 epochs. During this iterative process, the network simultaneously refines the wavefunction representation and tunes the energy parameter to obtain stable bound-state solutions for the $1s$, $1p$, and $1d$ orbitals.

3 Results and Discussion

The radial time-independent Schrödinger equation (TISE) was solved for the $1s$, $1p$, and $1d$ single-particle states of neutrons and protons using Physics-Informed Neural Networks (PINNs). The resulting bound-state energy eigenvalues are summarized in Table 1, alongside the corresponding experimental values reported in Ref. [5]. The comparison provides a quantitative assessment of the capability of PINNs to reproduce key features of nuclear single-particle spectra within a Woods-Saxon mean-field framework.

Table 1: Comparison between energies obtained using PINNs and the corresponding experimental values [5]

Particle	**State**	j	$\mathbf{E}_{PINNs}$ **(MeV)**	$\mathbf{E}_{exp}$ **(MeV) [5]**
Neutron	$1s$	$1/2$	-14.700	-14.668
Neutron	$1p$	$3/2$	-9.982	-10.068
Neutron	$1p$	$1/2$	-9.690	-10.068
Neutron	$1d$	$5/2$	-7.080	-7.368
Neutron	$1d$	$3/2$	-7.228	-7.368
Proton	$1s$	$1/2$	-10.504	-10.500
Proton	$1p$	$3/2$	-5.643	-6.200
Proton	$1p$	$1/2$	-5.539	-6.200
Proton	$1d$	$5/2$	-3.638	-3.800
Proton	$1d$	$3/2$	-2.499	-3.800

The ground-state $1s_{1/2}$ energies exhibit excellent agreement with experimental data for both neutrons and protons,

with deviations well below one percent. This result indicates that the PINN formulation is highly effective in capturing the dominant characteristics of the central Woods–Saxon potential and correctly identifying the lowest-energy bound solutions of the TISE. By embedding the governing differential equation directly into the loss function, the neural network converges to physically meaningful eigenvalues without reliance on spatial discretization or basis expansions.

Beyond the ground state, the PINN approach successfully reproduces the spin–orbit splitting observed in the $1p$ and $1d$ shells. In both neutron and proton spectra, states with total angular momentum $j = l + 1/2$ are consistently more deeply bound than their $j = l - 1/2$ counterparts. For example, the neutron $1p_{3/2}$ state is predicted at a lower energy than the $1p_{1/2}$ state, in agreement with established nuclear shell-model behavior. This demonstrates that the spin–orbit interaction term is properly incorporated and that automatic differentiation within the PINN framework accurately resolves the radial derivatives required to model this coupling.

A systematic energy shift is observed between neutron and proton states due to the inclusion of the Coulomb interaction for protons. While the central nuclear potential parameters are identical for both particles, proton energy levels are shifted toward higher energies as a result of electrostatic repulsion. For the $1s_{1/2}$ state, this Coulomb shift amounts to approximately 4.2 MeV, consistent with physical expectations. The stable convergence of the proton solutions indicates that the PINN methodology remains robust even in the presence of the additional Coulomb barrier, which introduces further nonlinearity into the potential.

For higher orbital states, particularly the $1d$ levels, the predicted eigenvalues follow the correct physical ordering and lie within the expected energy range, though with moderately larger deviations from experimental values. This behavior is commonly encountered in neural-network-based eigenvalue solvers, as excited-state wavefunctions possess additional nodes and more intricate spatial structures. Satisfying the Schrödinger residual across the entire radial domain therefore becomes increasingly challenging. Nevertheless, the agreement achieved for these states remains sufficient for qualitative and semi-quantitative nuclear structure analysis.

Overall, the results demonstrate that

Physics-Informed Neural Networks provide a reliable and physically consistent framework for solving the radial TISE in nuclear systems. The method captures essential features such as ground-state binding, spin–orbit splitting, and Coulomb effects, highlighting its potential as an alternative computational tool for nuclear mean-field calculations.

4 Conclusion

In this work, Physics-Informed Neural Networks (PINNs) were employed to solve the radial time-independent Schrödinger equation for neutron and proton single-particle states in the ^{40}Ca nucleus within a Woods–Saxon mean-field potential. By incorporating the governing quantum mechanical equations directly into the loss function, the PINN framework enables the determination of bound-state energy eigenvalues without reliance on spatial grids or predefined basis functions. The obtained results demonstrate that PINNs are capable of reproducing essential features of nuclear shell structure, including ground-state binding energies, spin–orbit splitting in the p and d shells, and Coulomb-induced energy shifts between neutron and proton states. The agreement with experimental energy levels confirms that the method provides physically consistent solutions across both low-lying and excited single-particle states. These findings indicate that PINNs offer a flexible and robust alternative to conventional numerical solvers used in nuclear structure calculations. Their mesh-free formulation and direct enforcement of physical constraints make them particularly attractive for problems involving complex potentials or extended parameter spaces. As a result, PINNs hold significant promise for future applications in nuclear mean-field modeling and related quantum many-body systems.

Acknowledgments

A. Awasthi acknowledges the financial support provided by the DST, Govt. of India, through Grant No. DST/INSPIRE Fellowship/2020/IF200538.

References

[1] Shikha Awasthi, OSKS Sastri, and Vandna Luthra. Semi-magic numbers from sub-shell closures in shell model energy levels. *Journal of Physics Education*, 39(1):1–22, 2025.

[2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[3] Kenneth S Krane. *Introductory nuclear physics*. John Wiley & Sons, 1991.

[4] Maziar Raissi, Paris Perdikaris, Nazanin Ahmadi, and George Em Karniadakis. Physics-informed neural networks and extensions. *arXiv preprint arXiv:2408.16806*, 2024.

[5] Aditi Sharma, Swapna Gora, Jithin Bhagavathi, and OSK Sastri. Simulation study of nuclear shell model using sine basis. *American Journal of Physics*, 88(7):576–585, 2020.

Solving Hydrogen Atom using Physics- Informed Neural Network for ground state

Kanika[1], **Ishwar Kant**[1], Ayushi Awasthi[1] , Arushi Sharma[1]
Department of Physical and Astronomical Sciences
Central University of Himachal Pradesh, HP 176215, Bharat(India)
ishwrknt@gmail.com

Submitted on 12-02-2026

Abstract

The hydrogen atom is a standard benchmark for testing and validating numerical techniques in quantum mechanics. Although high-precision studies in the literature include small quantum electrodynamic corrections such as the Lamb shift, the present work focuses on the numerical solution of the radial Schrödinger equation without incorporating these corrections.In this study, an unsupervised learning framework is employed to model radial wavefunctions at fixed energy values. A neural network is trained to learn physically acceptable solutions that satisfy boundary conditions and normalization behavior.
The obtained wavefunctions show strong agreement with expected theoretical profiles, reproducing correct nodal structures and asymptotic decay.The results demonstrate that unsupervised neural networks can serve as an effective numerical tool for approximating hydrogenic wavefunctions with good accuracy and precision. This approach provides a flexible foundation for future extensions toward more complex atomic systems and higher-order corrections.

1 Introduction

The hydrogen atom has long been a central model in quantum mechanics because it provides a clear and reliable framework for understanding atomic behavior. Since its exact analytical solution is known, it is often used to test the accuracy of numerical and computational methods.[1, 2] Studying the hydrogen atom also helps in analyzing subtle physical effects such as fine-structure corrections and the Lamb shift, which require highly precise calculations. For this reason, it remains an important system in modern computational physics.[3]

In earlier studies, researchers mainly relied on traditional numerical techniques to solve the radial Schrödinger equation. Methods such as finite-difference schemes and matrix approaches have been widely used[1] Among them, matrix methods with sine basis functions are popular because they provide stable eigenvalues and reason-

able wavefunctions. However, these methods depend on grid size and basis choice, which can sometimes reduce flexibility and accuracy, especially for excited states.[3]

More recently, Physics-Informed Neural Networks (PINNs) have emerged as a powerful approach for solving differential equations in quantum mechanics. By embedding physical laws and boundary conditions directly into the loss function, PINNs encourage solutions that remain smooth, stable, and physically consistent across the domain.[4, 5, 6] This physics-guided paradigm has gained considerable attention for integrating prior scientific knowledge into neural network training and for generating solutions that respect governing equations.[1, 7, 8]

Alongside these developments, an unupervised learning strategy is employed for the numerical simulation of the hydrogen atom.[9, 7] Rather than depending solely on physics-based constraints during optimization, the neural network is trained using Physics equations embedded in loss function.[10, 11, ?] This data-guided formulation enables stable learning of radial solutions and captures the essential structure of hydrogenic states.[12, 13] The predicted wavefunctions agree well with theoretical expectations, exhibiting correct nodal patterns and physically meaningful asymptotic decay. [1]Such an unsupervised framework improves accuracy, maintains physical consistency, and reduces numerical errors.[14, 15] This makes it a useful and efficient method for studying quantum eigenvalue problems like the hydrogen atom.[?, 16, 7]

2 Modeling of the Physical System

In this section, we elaborate the four stages of the modeling theory as suggested by Hestenes,15 viz., Description, Formulation, Ramification, and Validation, to clearly delineate various aspects of the model.[3]

2.1 Description Stage

This stage comprises three components:

2.1.1 Object Description

1. **Object Type:** Hydrogen atom , the simplest constitutent of matter .
2. **Object Composition:** An electron and a proton.
3. **Object Variables:** The electron has a mass $m_e = 0.5109 \times 10^6\,\mathrm{eV}/c^2$. The proton mass is approximately 1837 times larger than the electron mass, with $m_p = 938.272 \times 10^6\,\mathrm{eV}/c^2$. Both particles carry electric charges of equal magnitude, one elementary charge, but with opposite signs.

2.1.2 Interaction Description

1. **Type of Interaction:** Electromagnetic interaction between the proton and electron [3, 10].

2. **Agents of Interaction:**

   (i) Proton as Coulomb (Attractive) source

   (ii) Range of interaction ; typically extends infinitely long [3]

   Interaction Variables: The behaviour of the hydrogen atom depends on a few fundamental quantities. The electron–proton distance r determines how the interaction varies in space. The Coulomb factor $e^2/(4\pi\epsilon_0)$ sets the strength of the electric attraction between the two particles.

2.1.3 Process Description

(a) **Reference System:**

The hydrogen atom can be rigorously reduced to an effective single- particle problem by introducing the reduced mass of the electron-proton system:

$$\mu = \frac{m_p m_e}{m_p + m_e} \approx m_e, \tag{1}$$

where the approximation arises from $m_p \gg m_e$. The electron can thus be treated as a particle moving relative to the center-of-mass of the system.

As the inter-particle interaction depends solely on the radial separation, the system possesses central symmetry. Consequently, **spherical polar coordinates (SPC)** constitute the most appropriate reference frame for its analysis.[9, 3]

(b) **State Variables:**

The wavefunction $\psi(r,t)$ completely specifies the quantum state of the system, with all physical observables derivable from it through appropriate operators.

2.2 Formulation Stage

At this stage, we first discuss the dynamical laws that govern the motion of the proton and then specify the interaction laws in their mathematical form. The initial and boundary conditions, if any, are then specified to complete the formulation of the mathematical model representing the physical system.

2.2.1 Interaction Laws

The hydrogen atom is governed primarily by the Coulomb attraction between the electron and the proton, which provides the central potential responsible for binding the system, expressed as $V(r) = -\dfrac{e^2}{4\pi\epsilon_0 r}$. Beyond this dominant interaction, additional effects arise from relativistic and quantum electrodynamical contributions. [17].

2.2.2 Dynamical Laws

The time-dependent Schrödinger equation (TDSE)

$$i\hbar\frac{\partial\Psi(\mathbf{r},t)}{\partial t} = -\frac{\hbar^2}{2\mu}\nabla^2\Psi(\mathbf{r},t) + V(\mathbf{r})\Psi(\mathbf{r},t) \tag{2}$$

is transformed into the time-independent Schrödinger equation (TISE) using the

method of separation of variables technique ,

$$\Psi(\mathbf{r}, t) = \psi(\mathbf{r})\, T(t)$$

, one obtains

$$-\frac{\hbar^2}{2\mu}\nabla^2\psi(\mathbf{r}) + V(\mathbf{r})\psi(\mathbf{r}) = E\,\psi(\mathbf{r}). \quad (3)$$

The non-relativistic radial Schrödinger equation is:

$$\left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2\mu r^2} - \frac{e^2}{4\pi\epsilon_0 r}\right] u_{nl}(r) = E_{nl}u_{nl}(r). \quad (4)$$

2.2.3 Boundary Conditions

For a physically acceptable solution of the hydrogen atom, the radial wavefunction must satisfy well-defined boundary conditions. The wavefunction and its derivative are required to exist and remain continuous for all values of r, ensuring a finite and physically meaningful solution. Dirichlet boundary conditions are imposed such that $u_{nl}(0) = 0$ and $u_{nl}(L) = 0$, which reflect the regular behaviour at the origin and the decay of the wavefunction at large distances. Additionally, the wavefunction must be normalisable so that the total probability of finding the electron in space remains finite.

2.3 Ramification Stage

An **unsupervised-learning-based Physics-Informed Neural Network (PINN)** is employed to numerically obtain hydrogenic radial wavefunctions at fixed energies while satisfying physical boundary conditions.

2.3.1 PINNs with Boundary, Normalization, and Orthogonality Constraints

A **Physics-Informed Neural Network** is employed to obtain the solution. The network takes the radial coordinate r as input and outputs the corresponding wavefunction $\psi(r)$.

The total loss function of the PINN is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{norm}} + \mathcal{L}_{\text{orth}}, \quad (5)$$

where:

(i) **Physics loss ($\mathcal{L}_{\text{physics}}$)**: Ensures the Schrödinger equation with is satisfied:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_r}\sum_{i=1}^{N_r}\left|\hat{H}\psi(r_i) - E\psi(r_i)\right|^2 \quad (6)$$

(ii) **Boundary condition loss ($\mathcal{L}_{\text{BC}}$)**: Enforces $\psi(0) = 0$ and $\psi(L) = 0$.

(iii) **Normalization loss ($\mathcal{L}_{\text{norm}}$)**: Ensures

$$\int_0^L |\psi(r)|^2\, dr = 1 \quad (7)$$

(iv) **Orthogonality loss ($\mathcal{L}_{\text{orth}}$)**: Ensures orthogonality of different eigenstates:

$$\int_0^L \psi_m(r)\,\psi_n(r)\, dr = 0 \quad (m \neq n) \quad (8)$$

2.3.2 Optimization Workflow

The overall workflow for the ramification stage is as follows:

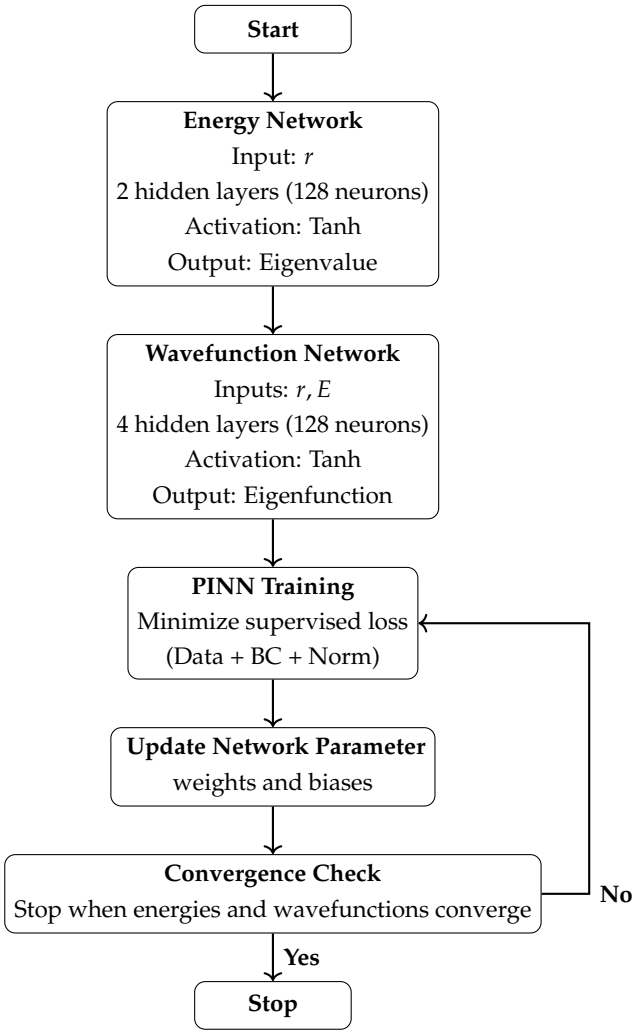


Figure 1: Optimization workflow.

2.4 Validation Stage

(i) Comparison can be made between theoretical and numerically Obtained energy eigenvalues and eigen functions from our simulation .

(ii) Finally, the energy level sequence that we obtain must also match with that expected from experimental considerations, so as to validate the deduced configurations.

3 Numerical Simulation

3.1 Choice of Units

All simulations are performed in atomic units ($\hbar = m_e = e = 1$), in which energies are measured in Hartree and lengths in Bohr radii. In this unit system, the electron–proton interaction reduces to the standard Coulomb form

$$V(r) = -\frac{1}{r},$$

and the exact nonrelativistic ground-state energy

$$E_{1s} = -0.5\ \text{Ha}$$

serves as a precise reference for evaluating numerical convergence.[3]

For higher orbital angular momentum, the effective potential is typically written as

$$V_{\text{eff}}(r) = -\frac{1}{r} + \frac{l(l+1)}{2r^2}, \tag{9}$$

highlighting its explicit l-dependence through the centrifugal term [3].

In the literature, Lamb-shift–related QED corrections are commonly treated by retaining the fine-structure constant α explicitly and extending the Hamiltonian with terms such as vacuum-polarization, Darwin, and self-energy contributions to enable stable evaluation of radiative level shifts.

3.2 Region of Interest

The radial coordinate is confined to a finite domain $r \in [0, R_{\max}]$, where $R_{\max}$ is chosen sufficiently large to ensure that bound-state wavefunctions decay to numerically negligible values at the boundary. For Physics-Informed Neural Network method , typical choices such as $R_{\max} = 20$–45 Bohr sufficient to produce hydrogenic energies upto atleast n=2 with accuracy . For principal quantum numbers $n \geq 3$, the radial cutoff is typically chosen in the range $R_{\max} = 80$–100 Bohr.[3]

3.3 Choice of Numerical technique

Accurate modelling of the hydrogen atom requires a method that remains stable near the Coulomb singularity and reliable for excited states. Conventional approaches such as finite-difference schemes and standalone basis expansions often demand fine discretization and careful boundary treatment. These requirements increase computational cost and can introduce numerical errors[18]. To address these issues, Physics-Informed Neural Networks (PINNs) using unsupervised learning was adopted. PINNs incorporate the governing equation and boundary conditions directly into the training process, reducing dependence on dense meshes[1]. This combination improves numerical stability and overall accuracy. It also produces smooth and physically consistent wavefunctions. This approach balances efficiency with precision. As a result, it serves as a practical alternative to traditional single-method techniques.

4 Results an Discussion

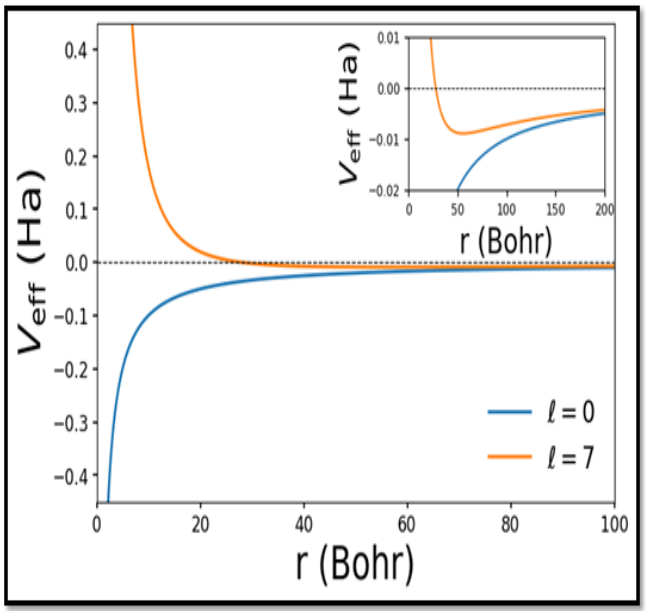


Figure 2: Effective Potential of Hydrogen atom for $l = 0$ and $l = 7$

Acknowledgments

A. Awasthi acknowledges the financial support provided by the DST, Govt. of India, through Grant No. DST-INSPIRE fellowship/2020/IF200538.

References

[1] Henry Jin, Pavlos Protopapas, and Marios Mattheakis. Physics-informed neural networks for quantum eigenvalue problems. 2022.

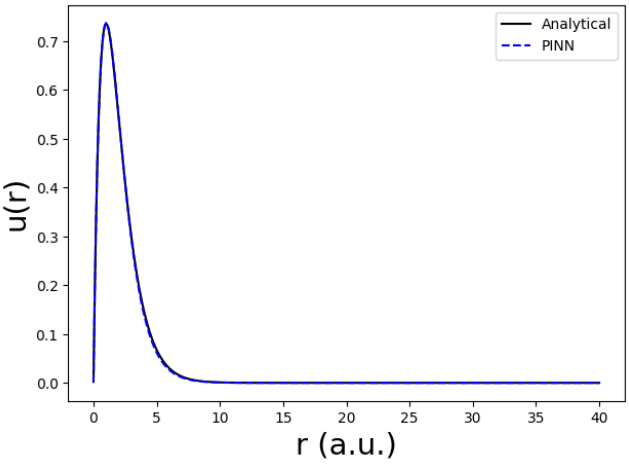


Figure 3: Comparison of analytical and PINN-predicted radial wavefunctions for the 1s state ($n = 1$, $l = 0$), achieving an accuracy of 2.576×10^{-4}.

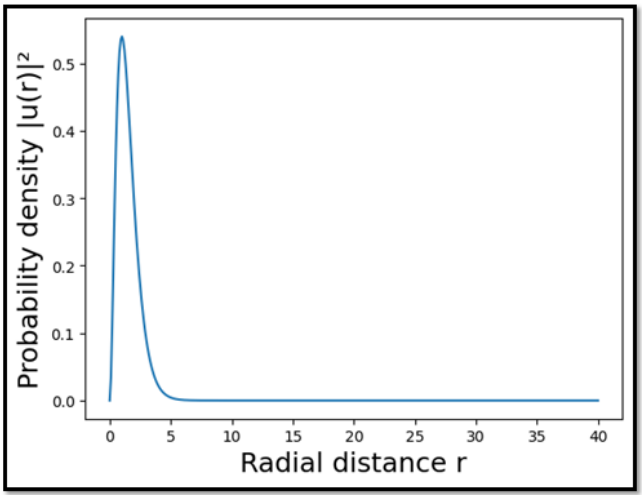


Figure 4: PINN-predicted Probability Density for the 1s state ($n = 1$, $l = 0$).

[2] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. Cambridge University Press, 3rd edition, 2018.

[3] O. S. K. S. Sastri. *Computer Simulations in Quantum Physics Using Numerical Techniques, Worksheets and Programming*. 2024. Book Series towards Implementation of NEP-2020.

[4] Hubert Baty. Solving stiff ordinary differential equations using physics-informed neural networks (pinns): Simple recipes to improve training of vanilla-pinns. 2023.

[5] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. 2018.

[6] Hubert Baty and Léo Baty. Solving differential equations using physics-informed deep learning: A hands-on tutorial with benchmark tests. 2023. Preprint, dated April 5, 2023.

[7] S. Desai, M. Mattheakis, H. Joy, P. Protopapas, and S. Roberts. One-shot transfer learning of physics-informed neural networks, 2021.

[8] Michael Magill, Faisal Qureshi, and Hendrik W. de Haan. Neural networks trained to solve differential equations learn general representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[9] Salvatore Cuomo, Vito S. Di Cola, Ferdinando Giampaolo, Gianluigi Rozza, Maziar Raissi, and Fabio Piccialli. Solving differential problems with physics-informed neural networks: A review

of recent methods and advances in scientific computing. *Journal of Scientific Computing*, 92:88, 2022.

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks. *Journal of Computational Physics*, 378:686–707, 2019.

[11] Henry Jin, Marios Mattheakis, and Pavlos Protopapas. Unsupervised neural networks for quantum eigenvalue problems. 2020.

[12] Marios Mattheakis, Pavlos Protopapas, David Sondak, Matteo Di Giovanni, and Efthimios Kaxiras. Physical symmetries embedded in neural networks, 2020.

[13] Cedric Flamant, Pavlos Protopapas, and David Sondak. Solving differential equations using neural network solution bundles. 2020.

[14] Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[15] Marios Mattheakis, David Sondak, A. S. Dogra, and Pavlos Protopapas. Hamiltonian neural networks for solving differential equations, 2020.

[16] George E. Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.

[17] M. Lieber. Symmetry of the hydrogen atom and the lamb shift. *Physical Review*, 174(5):2037, 1968.

[18] O. S. K. S. Sastri et al. Numerical solution of square well potential with matrix method using worksheets. *Physics Education*, 36:1–14, 2019.