Physics-Informed Neural Network (PINNs) Solution of One-Dimensional Steady-State Heat Conduction in a homogeneous rod

Arya Abdollahi
Department of Mechanical Engineering
Iran University of Science and Technology
16846-13114, Tehran, Iran
arya.abdollahi.t@gmail.com

November 26, 2025

Abstract

This report presents a physics-informed neural networks (PINNs) formulation for solving a one-dimensional steady-state heat conduction problem in a homogeneous rod. Starting from Fourier's law of heat conduction and an energy balance over a differential control volume, the governing equations are derived and simplified under the assumptions of constant thermal conductivity, no internal heat generation, and steady one-dimensional conduction. The resulting boundary value problem is then solved using a fully-connected neural network whose training is guided by the residual of the differential equation and the imposed boundary conditions, rather than by labeled data. Automatic differentiation is used to evaluate the required spatial derivatives of the temperature field. The performance of the PINN is evaluated by comparing the predicted temperature distribution with the analytical solution. Numerical experiments demonstrate excellent agreement, with relative errors on the order of 10^{-6}, confirming that the PINN can accurately capture the underlying physics of the problem.

Contents

1 Introduction

The numerical solution of partial differential equations (PDEs) has traditionally relied on classical computational methods such as the finite difference method (FDM), finite volume method (FVM), and finite element method (FEM). In the context of heat transfer and fluid mechanics, these approaches are the foundation of computational fluid dynamics (CFD), where the physical domain is discretized into a grid and the governing equations are transformed into large systems of algebraic equations. Although highly successful and widely used in industry and academia, conventional CFD methods may face challenges when dealing with complex geometries, moving boundaries, multi-physics coupling, or when high-resolution meshes lead to prohibitive computational costs. Furthermore, extending classical methods to inverse problems or data-driven settings often requires nontrivial modifications.

Physics-Informed Neural Networks (PINNs) have emerged as a promising alternative paradigm that integrates deep learning with the governing laws of physics [1]. Instead of learning purely from data, a PINN incorporates the residual of the governing PDEs, boundary conditions, and any available measurements directly into its loss function [1, 2]. The neural network acts as a universal function approximator for the solution field, and automatic differentiation is used to compute the spatial and temporal derivatives required by the PDE. This eliminates the need for grid generation and facilitates the handling of irregular geometries or scattered data. Moreover, the same framework can naturally address both forward problems (predicting the solution given parameters) and inverse problems (estimating parameters or unknown boundary conditions from partial observations) [2].

Recent literature demonstrates PINNs being applied to a wide range of problems, including unsteady fluid flows governed by the Navier–Stokes equations, transport phenomena, structural mechanics, and multi-physics systems [2]. Applications span from modeling blood flow in biomedical engineering, to turbulence modeling in aerodynamics, to uncertainty quantification and parameter inference in energy systems. For high-dimensional problems or domains with limited data, PINNs offer an appealing combination of physical consistency and flexibility, especially when leveraged with modern GPU hardware.

In this project, we focus on a fundamental problem in heat transfer: one-dimensional steady-state heat conduction in a homogeneous rod. Although the problem is simple and admits a closed-form analytical solution [3], it serves as a clean testbed to illustrate the core ideas of the PINN methodology. The physical setting is a rod of length $L = 2.5\,\mathrm{m}$, with prescribed temperatures at its two ends. Heat is conducted along the rod in the absence of internal heat generation. The goal is to predict the temperature distribution along the rod. Figure 1 shows a schematic of the physical domain, where the left end of the rod is held at constant temperature $T_0 = 100°\text{C}$ and the right end at $T_1 = 300°\text{C}$. Under the assumptions of constant thermal conductivity and steady, one-dimensional conduction, the governing equation is derived from Fourier's law of heat transfer and an energy balance over a differential control volume [3]. These derivations are detailed in Section 2. The resulting boundary value problem is then reformulated in nondimensional form and solved using a PINN that approximates the temperature field θ as a function of a nondimensional spatial coordinate ξ.

The rest of this report is organized as follows. Section 2 derives the mathematical model, starting from Fourier's law and the general heat conduction equation. Section 3 describes the PINN architecture and the construction of the physics-based loss function. Section 4 presents numerical results and compares the PINN solution with the analytical solution. Finally, Section 5 summarizes the findings and outlines possible extensions of this work.

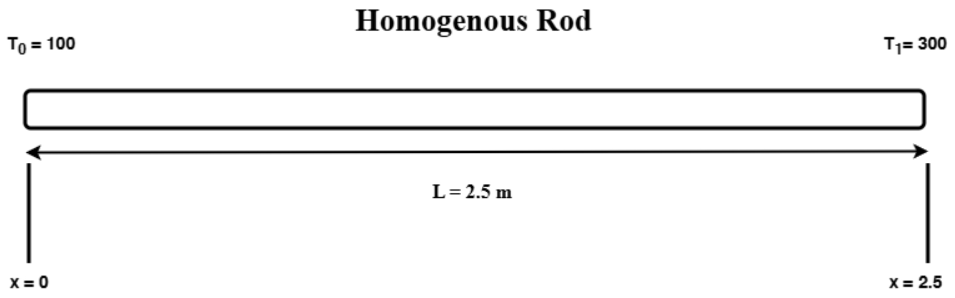

Figure 1: Schematic of physical problem.

2 Mathematical Formulation

In this section, we derive the governing equations for one-dimensional steady-state heat conduction in a homogeneous rod, starting from Fourier's law of heat transfer and an energy balance over a differential control volume. Step by step, we introduce the necessary assumptions and simplify the general heat conduction equation to its final one-dimensional form.

2.1 Fourier's Law of Heat Conduction

Fourier's law states that the heat flux vector is proportional to the negative gradient of temperature. In three dimensions, the conductive heat flux vector $\boldsymbol{q}$ is given by

$$\boldsymbol{q} = -k\nabla T, \tag{1}$$

where k is the thermal conductivity of the material (assumed isotropic and, for now, possibly independent on temperature and position leading to constant k), and $T = T(\boldsymbol{x}, t)$ is the fucntion which resembles the distribution of temperature on time and space. The negative sign reflects the fact that heat flows from regions of higher temperature to regions of lower temperature.

In component form, Equation (1) can be written as

$$\boldsymbol{q_x} = -k\frac{\partial T}{\partial x}, \qquad \boldsymbol{q_y} = -k\frac{\partial T}{\partial y}, \qquad \boldsymbol{q_z} = -k\frac{\partial T}{\partial z}, \tag{2}$$

where $\boldsymbol{q_x}$, $\boldsymbol{q_y}$, and $\boldsymbol{q_z}$ are the components of the heat flux vector in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ directions, respectively.

$$\vec{\boldsymbol{q}} = \boldsymbol{q_x}\,\hat{\imath} + \boldsymbol{q_y}\,\hat{\jmath} + \boldsymbol{q_z}\,\hat{k} \tag{3}$$

The vector form of Fourier's law is particularly valuable when analyzing heat conduction in multiple dimensions, as it compactly represents the relationship between temperature gradients and the resulting heat flux in all spatial directions. Equation (3) will be used later to develop the general heat conduction equation.

2.2 Energy Balance over a Differential Control Volume

Consider a differential control volume of size $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ fixed in space. The general transient heat conduction equation with possible internal heat generation $\dot{q}'''$ (per unit volume) is obtained by applying conservation of energy:

$$\begin{aligned}\text{(Rate of increase of thermal energy inside CV)} = {} & \text{(Net rate of heat conduction into CV)} \\ & + \text{(Rate of internal heat generation)}\end{aligned} \tag{4}$$

The rate of increase of thermal energy per unit volume is given by $\rho c_p \frac{\partial T}{\partial t}$, where ρ is the density and c_p is the specific heat capacity at constant pressure. Thus, the total rate of thermal energy increase inside the control volume is

$$\rho c_p \frac{\partial T}{\partial t}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \tag{5}$$

The net rate of heat conduction into the control volume can be expressed using the divergence of the heat flux:

$$-\nabla \cdot \boldsymbol{q} \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z \tag{6}$$

where the negative sign arises because a positive divergence of $\boldsymbol{q}$ corresponds to net heat leaving the control volume. Using Fourier's law, $\boldsymbol{q} = -k\nabla T$, we have

$$-\nabla \cdot \boldsymbol{q} = -\nabla \cdot (-k\nabla T) = \nabla \cdot (k\nabla T) \tag{7}$$

The rate of internal heat generation per unit volume is $\dot{q}'''$, so the total rate of internal heat generation inside the control volume is

$$\dot{q}''' \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z \tag{8}$$

Substituting these expressions into the energy balance in Equation (4), and dividing by the volume element, yields the general heat conduction equation:

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T) + \dot{q}''' \tag{9}$$

2.3 Assumptions for the 1D Steady Conduction Problem

For the specific problem considered in this project, we make the following assumptions:

1. **Steady state:** The temperature does not vary with time, so
$$\frac{\partial T}{\partial t} = 0 \tag{10}$$
2. **One-dimensional conduction:** Heat transfer occurs only along the x-direction (along the length of the rod). There are no temperature gradients in the y or z directions:
$$\frac{\partial T}{\partial y} = 0, \qquad \frac{\partial T}{\partial z} = 0 \tag{11}$$
3. **Constant thermal conductivity:** The thermal conductivity k is constant throughout the rod and does not depend on temperature or position.
4. **No internal heat generation:** There is no volumetric heat source within the rod, so
$$\dot{q}''' = 0 \tag{12}$$

Applying these assumptions to Equation (9), we first drop the transient term due to steady state:

$$0 = \nabla \cdot (k\nabla T) + \dot{q}''' \tag{13}$$

With $\dot{q}''' = 0$, this reduces to

$$0 = \nabla \cdot (k \nabla T) \tag{14}$$

For constant k, the thermal conductivity can be taken outside the divergence operator:

$$0 = k \nabla^2 T, \tag{15}$$

or equivalently

$$\nabla^2 T = 0 \tag{16}$$

Under the one-dimensional assumption, the Laplacian simplifies to

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2}, \tag{17}$$

so the governing equation becomes

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = 0 \tag{18}$$

2.4 Boundary Conditions and Nondimensionalization

We consider a rod of length L, with prescribed temperatures at the two ends:

$$T(0) = T_0, \tag{19}$$

$$T(L) = T_1 \tag{20}$$

With the governing equation and boundary conditions identified, we proceed by nondimensionalizing the spatial variable and temperature. This step enhances interpretability and improves numerical treatment of the problem.

$$\xi = \frac{x}{L}, \qquad \theta(\xi) = \frac{T(x) - T_0}{T_1 - T_0} \tag{21}$$

Using the chain rule,

$$\frac{\mathrm{d}T}{\mathrm{d}x} = \frac{\mathrm{d}T}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x} = \frac{1}{L}\frac{\mathrm{d}T}{\mathrm{d}\xi}, \tag{22}$$

and

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{L}\frac{\mathrm{d}T}{\mathrm{d}\xi}\right) = \frac{1}{L^2}\frac{\mathrm{d}^2 T}{\mathrm{d}\xi^2} \tag{23}$$

Substituting into Equation (18) gives

$$\frac{1}{L^2}\frac{\mathrm{d}^2 T}{\mathrm{d}\xi^2} = 0 \quad \Rightarrow \quad \frac{\mathrm{d}^2 T}{\mathrm{d}\xi^2} = 0 \tag{24}$$

Using the definition of θ, we have

$$T(x) = T_0 + (T_1 - T_0)\theta(\xi), \tag{25}$$

so

$$\frac{\mathrm{d}^2 T}{\mathrm{d}\xi^2} = (T_1 - T_0)\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} \tag{26}$$

Substituting into the previous expression and noting that $T_1 - T_0 \neq 0$, we obtain the nondimensional governing equation

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} = 0, \tag{27}$$

with corresponding boundary conditions

$$\theta(0) = 0, \qquad \theta(1) = 1 \tag{28}$$

This is the boundary value problem that will be solved using a Physics-Informed Neural Network in the subsequent sections.

3 PINN Formulation

Physics-Informed Neural Networks (PINNs) provide a flexible framework for solving differential equations by embedding the governing physical laws directly into the loss function of a neural network. Instead of relying on discretized numerical schemes or labeled data, PINNs enforce the residuals of the differential equations and boundary conditions through automatic differentiation. In this section, we describe the construction of the PINN used to approximate the non-dimensional temperature field $\theta(\xi)$ for the one-dimensional steady-state heat conduction problem derived in Section 2.

3.1 Neural Network Approximation

The solution $\theta(\xi)$ is approximated by a fully-connected feedforward neural network,

$$\theta_{\mathrm{NN}}(\xi;\, \mathbf{p}),$$

where $\mathbf{p}$ denotes the set of trainable parameters of neural network (weights and biases). The network used in this work consists of several hidden layers with tanh activation function, chosen for their smoothness and compatibility with automatic differentiation. The network takes the nondimensional spatial coordinate $\xi \in [0, 1]$ as input and outputs a scalar temperature value. Through training, the PINN learns a mapping that satisfies the governing differential equation and boundary conditions without requiring any direct temperature measurements.

3.2 Automatic Differentiation and PDE Residual

A central advantage of PINNs is the use of automatic differentiation to compute the required temperature gradients. For the present problem, the nondimensional governing equation is

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\xi^2} = 0$$

Using automatic differentiation, the second derivative of the neural network output can be evaluated exactly:

$$\theta_\xi = \frac{\mathrm{d}\theta_{\mathrm{NN}}}{\mathrm{d}\xi}, \qquad \theta_{\xi\xi} = \frac{\mathrm{d}^2\theta_{\mathrm{NN}}}{\mathrm{d}\xi^2}$$

The PDE residual at any collocation point $\xi_f^{(i)}$ is then defined as

$$r_{\mathrm{PDE}}(\xi_f^{(i)}) = \theta_{\xi\xi}(\xi_f^{(i)}),$$

which should vanish for the exact solution. Interior collocation points are sampled uniformly in the interval $(0, 1)$, and the PINN is trained to minimize the squared residual at these points.

3.3 Boundary Condition Enforcement

The nondimensional boundary conditions for the problem are

$$\theta(0) = 0, \qquad \theta(1) = 1$$

These are enforced by defining boundary losses,

$$r_{\mathrm{BC},0} = \theta_{\mathrm{NN}}(0), \qquad r_{\mathrm{BC},1} = \theta_{\mathrm{NN}}(1) - 1$$

Unlike classical numerical methods that impose boundary conditions through node values or flux constraints, PINNs incorporate boundary information through penalty terms in the loss function. This makes the network train toward satisfying both the PDE and the boundary conditions simultaneously.

3.4 Loss Function

The total loss function combines the PDE residual and the boundary condition penalties:

$$\mathcal{L} = \mathcal{L}_{\mathrm{PDE}} + \mathcal{L}_{\mathrm{BC}},$$

where

$$\mathcal{L}_{\mathrm{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left(r_{\mathrm{PDE}}(\xi_f^{(i)}) \right)^2 ,$$

and

$$\mathcal{L}_{\mathrm{BC}} = \frac{1}{2} \left(r_{\mathrm{BC},0}^2 + r_{\mathrm{BC},1}^2 \right)$$

Here, N_f denotes the number of interior collocation points. The network parameters are optimized using a two-stage procedure: an initial training phase with the Adam optimizer for rapid gradient-based updates, followed by a refinement with the L-BFGS algorithm to achieve high accuracy and smooth convergence.

3.5 Summary of the PINN Procedure

The overall PINN methodology for this problem can be summarized as follows:

1. Sample interior collocation points $\xi_f^{(i)} \in (0, 1)$.
2. Evaluate $\theta_{\mathrm{NN}}(\xi)$ and compute θ_ξ and $\theta_{\xi\xi}$ using automatic differentiation.
3. Form PDE and boundary residuals.
4. Minimize the combined loss function $\mathcal{L}$ using gradient-based optimization.
5. Compare the trained PINN solution with the analytical solution for validation.

This formulation allows the neural network to approximate the temperature distribution while inherently respecting the physics of the problem. The numerical results and comparison with the analytical solution are presented in Section 4.

4 Results and Discussion

In this section, the performance of the Physics-Informed Neural Network (PINN) for solving the nondimensional and dimensional temperature fields is evaluated. The PINN predictions are compared with the analytical solution for both the non-dimensional temperature $\theta(\xi)$ and the dimensional temperature $T(x)$. Additional error plots are included to assess the accuracy, stability, and general behaviour of the trained model across the spatial domain.

4.1 Nondimensional Temperature Distribution

Figure 2 shows the comparison between the analytical solution $\theta_{\mathrm{exact}}(\xi) = \xi$ and the PINN-predicted solution. The two curves are almost indistinguishable, indicating that the neural network successfully approximates the linear temperature distribution expected for steady one-dimensional conduction. The close overlap demonstrates that the PINN has effectively enforced both the governing physics and the boundary conditions.

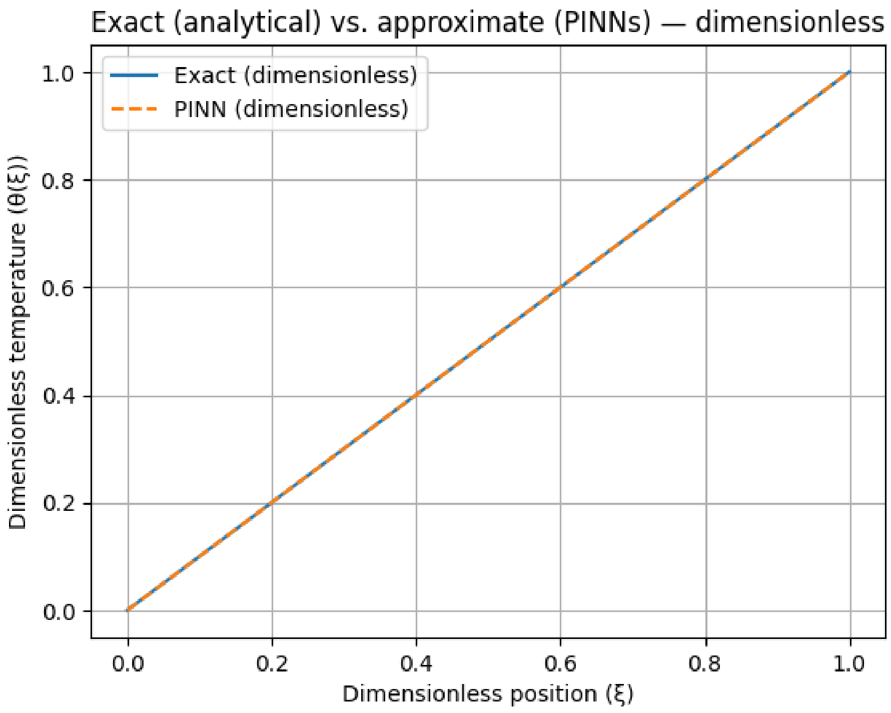

Figure 2: Comparison between analytical and PINN-predicted nondimensional temperature distribution

To further quantify model performance, the absolute error between the PINN prediction and the analytical solution is shown in Figure 3. The error remains on the order of 10^{-5} throughout the domain, demonstrating excellent agreement and numerical stability. The oscillatory pattern in the error distribution is expected, as the neural network approximates the solution as a smooth nonlinear mapping. This pattern reflects the interplay between the activation functions, network architecture, and optimizer but does not indicate any significant deviation from the true solution. The next phase of investigation could examine how changing the configuration of the hidden layers in this neural network influences its absolute error behavior throughout the parameters.

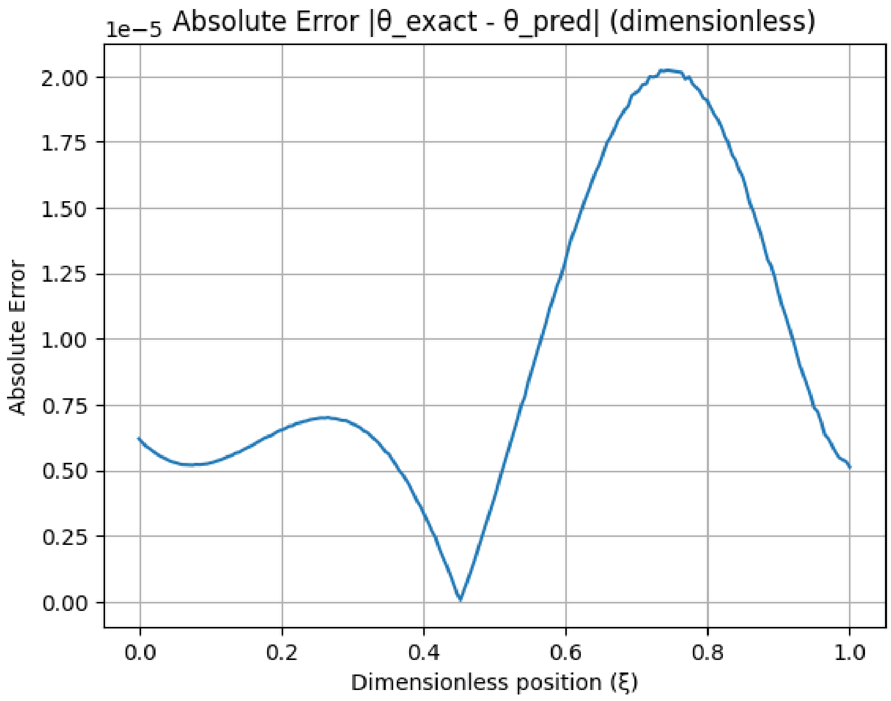

Figure 3: Absolute error in nondimensional temperature: $|\theta_{\text{exact}} - \theta_{\text{PINN}}|$

4.2 Dimensional Temperature Distribution

After obtaining the nonßdimensional solution, the dimensional temperature field was reconstructed using

$$T(x) = T_0 + (T_1 - T_0)\,\theta$$

Figure 4 illustrates the comparison between the analytical dimensional temperature and the PINN prediction. Similar to the non-dimensional case, the PINN solution aligns closely with the analytical linear temperature profile. The increased temperature range (from 100°C to 300°C) does not degrade network accuracy, confirming that the non-dimensional formulation effectively stabilizes the learning process.

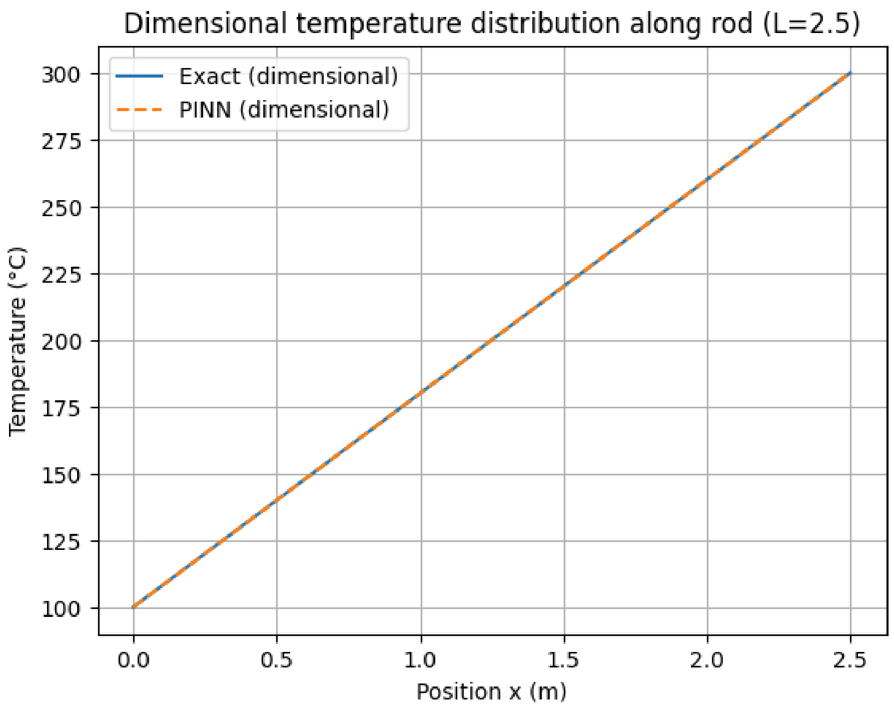

Figure 4: Dimensional temperature distribution along the rod for $L = 2.5$.

To assess local accuracy, the absolute difference between the analytical and PINN-predicted dimensional temperatures is shown in Figure 5. The error remains below 5×10^{-3} across the entire rod, which corresponds to less than 0.5% of the temperature span. The error pattern mirrors that of the nondimensional case, confirming that nondimensional transformation and subsequent reconstruction do not introduce new sources of numerical error.

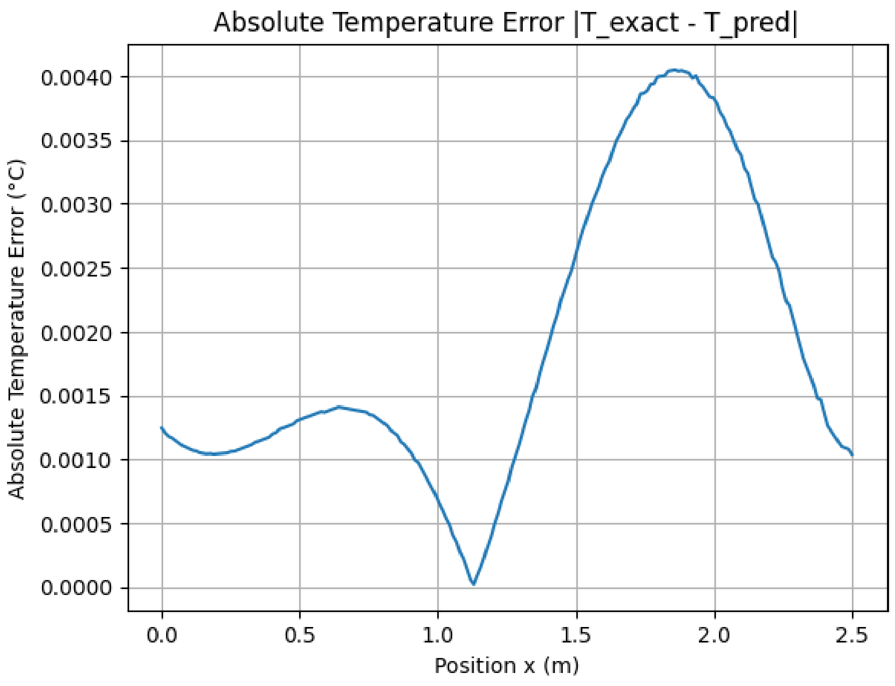

Figure 5: Absolute dimensional temperature error: $|T_{\text{exact}} - T_{\text{PINN}}|$.

4.3 Discussion and Validation

Overall, the PINN shows excellent predictive accuracy for both non-dimensional and dimensional formulations of the heat conduction problem. The nearly perfect overlap between analytical and PINN predictions confirms that the embedded physics and boundary conditions are enforced correctly. It is necessary to emphasize that we did not use classical CFD methods such as grid generation to solve this problem. Therefore, the PINNs method has no theoretical dependence on the previously mentioned method. The very small magnitude of the residual errors validates the effectiveness of the training procedure, especially the combination of Adam and L-BFGS optimizers.

The oscillatory structure observed in the error plots is typical of PINN-based solutions due to the nonlinear nature of neural networks and the smoothness imposed by tanh activations. These oscillations do not indicate instability; rather, they reflect the manner in which the neural network interpolates the solution space. Importantly, the amplitude of these oscillations is extremely small relative to the true temperature range.

Validation against the analytical solution confirms that the PINN accurately solves the boundary value problem and maintains physical consistency across the domain. The non-dimensionalization approach contributes significantly to numerical stability and allows the network to converge efficiently toward the analytical profile. These results demonstrate that PINNs are fully capable of solving simple boundary value problems with high accuracy, serving as a foundation for tackling more complex heat transfer problems in future work.

5 Conclusion

In this work, a Physics-Informed Neural Network (PINN) was developed and applied to the one-dimensional steady-state heat conduction problem in a homogeneous rod. The governing differential equation was derived from first principles using Fourier's law of heat conduction and an energy balance over a differential control volume. Through non-dimensionalization, the problem was reformulated in a general form that enhanced numerical stability and simplified the training process.

The PINN was constructed as a fully connected feedforward neural network and trained using a composite loss function that enforced both the governing physics and the boundary conditions through automatic differentiation. By sampling interior collocation points and evaluating the differential equation residual directly, the network was able to approximate the analytical linear temperature distribution without the need for any labeled data or discretized numerical mesh.

The resulting predictions demonstrated excellent agreement with the analytical solution in both non-dimensional and dimensional forms. The absolute errors remained small throughout the domain, confirming the accuracy and stability of the PINN approach. The results further showed that non-dimensionalization aids in improving numerical conditioning, and the combined use of Adam and L-BFGS optimizers produces a robust training strategy for this class of problems.

Overall, this project demonstrates that PINNs provide an effective and mesh-free alternative to traditional numerical methods for solving boundary value problems in heat transfer. The methodology is flexible, easily generalizable, and capable of delivering high-fidelity solutions. These results lay the groundwork for extending the PINN framework to more complex scenarios, including **internal heat generation**, **nonlinear thermal conductivity**, **multidimensional domains**, and **transient thermal** problems. Future work may also explore inverse formulations, parameter estimation, and coupling with fluid flow for convection–diffusion systems, further highlighting the versatility and potential of physics-informed machine learning in computational heat transfer.

Appendices

A Neural Network Architecture

The PINN used in this study is a fully-connected multilayer perceptron (MLP). The network consists of an input layer accepting the non-dimensional coordinate ξ, multiple hidden layers with tanh activation functions, and a single output neuron representing the nondimensional temperature $\theta(\xi)$.

Figure 6 gives a conceptual illustration of the network architecture, showing the layer connectivity and activation flow. This diagram is for structural reference and is not drawn to scale.

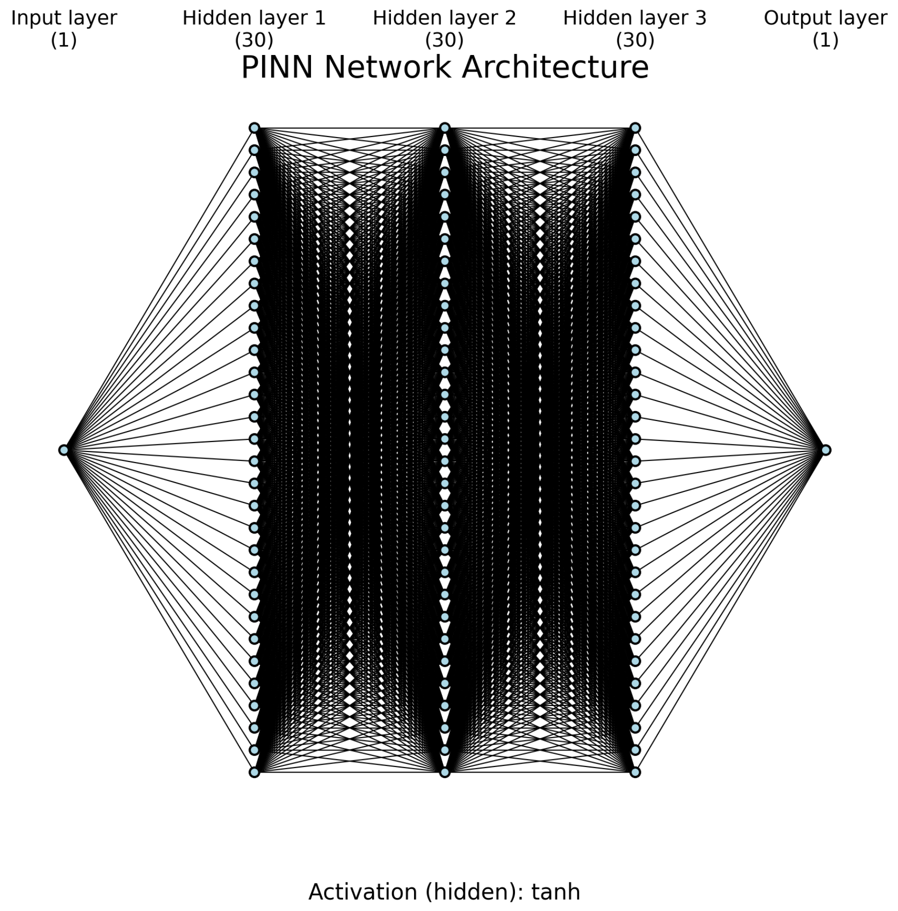

Figure 6: Conceptual architecture of the Physics-Informed Neural Network.

B Training Hyperparameters

The following hyperparameters were used to train the PINN for the one-dimensional heat conduction problem:

- **Network structure:** 1 input neuron, 3 hidden layers with 30 neurons each, 1 output neuron.
- **Activation function:** Hyperbolic tangent (tanh) for all hidden layers.
- **Optimizer (Phase 1):** Adam with learning rate 10^{-3}, trained for 5000 iterations.
- **Optimizer (Phase 2):** L-BFGS quasi-Newton optimizer with strong Wolfe line search.
- **Physics collocation points:** $N_f = 200$ uniformly sampled in $(0, 1)$.
- **Boundary points:** $\xi = 0$ and $\xi = 1$, enforced through Dirichlet boundary condition.
- **Loss weighting:** Equal weighting between PDE loss and boundary condition loss.

These hyperparameters were found to reliably produce stable and accurate solutions for the steady-state heat conduction problem.

C Code Summary

This appendix summarizes the software structure and major components used to implement the PINN:

C.1 Python Modules

- `models/fully_connected_pinn.py` Defines the neural network architecture and activation functions.
- `physics/heat1d_pde.py` Implements automatic differentiation to compute $\theta''(\xi)$.
- `physics/boundary_conditions.py` Implements Dirichlet boundary residuals at $\xi = 0$ and $\xi = 1$.
- `training/trainer.py` Constructs the combined loss function and executes training iterations.
- `utils/data_sampling.py` Generates collocation points in the spatial domain.
- `utils/plotting.py` Produces nondimensional and dimensional plots for validation.
- `main_project1.py` The main driver script responsible for model creation, training, evaluation, and saving results.

C.2 Execution Workflow

1. Load configuration parameters from `configs/config.yaml`.
2. Generate collocation and boundary points.
3. Construct the PINN model and initialize weights.
4. Train using Adam followed by L-BFGS.
5. Evaluate the trained PINN on a dense grid.
6. Compare with analytical solution and generate plots.
7. Save numerical outputs and figures.

C.3 Source Code Repository

All scripts, configuration files, trained models, and report materials used in this project are openly available at:

https://github.com/the-AryaAB/1.
-1D-Steady-Heat-Conduction-in-a-Rod-Using-PINNs

This repository contains the full folder structure, including:

- `src/` — model, physics, training, and utilities
- `docs/` — this report and figures
- `experiments/` — saved results and saved models
- `src/configs/` — YAML configuration file

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. DOI: 10.1016/j.jcp.2018.10.045

[2] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021. DOI: 10.1038/s42254-021-00314-5

[3] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 7th ed., Wiley, 2011. DOI: 10.1002/9780470549913