Physics-Informed Neural Networks for Full-Field Transient Temperature Reconstruction in Dimension-Reduced Fin Systems with Sparse Data

Md. Muminul Islam
Independent Researcher
Dhaka, Bangladesh
ORCID: 0009-0003-5463-8771

July 2026

Abstract

Heat transfer through extended surfaces (fins) is widely used in engineering applications ranging from portable electronic devices to spacecraft to improve heat dissipation. Accurate prediction of the temperature distribution along the fins is essential to optimize them for desired applications. This study presents a Physics-Informed Neural Network (PINN) approach to solve the transient fin equation.

In this study, the neural network predicts the temperature distribution by satisfying the governing physical laws and the sparse temperature sensor data, representing the most practical applications where embedded sensor probes gather data and feed it to the appropriate controller to drive the device.

This study emphasizes the use of PINN rather than conventional numerical methods, which need spatial discretization of the computational domain through mesh generation. The prediction accuracy in numerical methods strongly depends on the optimization of the quality and resolution of the mesh. The mesh optimization algorithm in a numerical solver plays an important role in mesh quality. This approach can increase the computational cost for complex geometries and is not suitable for real-time heat tracking and optimization.

In contrast, PINNs employ a neural network to estimate the solution function from the governing physical laws and to reduce the dependency on conventional mesh-based discretization. Furthermore, PINNs have the potential for real-time heat tracking and optimization, thanks to modern neural processing units (NPUs). Training PINNs can be computationally expensive; however, this study presents a novel approach to developing lightweight PINN models that can be trained effectively using limited sparse datasets.

This study developed a three-dimensional straight transient fin model to generate numerical simulation data and considered it as a reference dataset. The three-dimensional model was simplified to a one-dimensional model for the PINN approach. The PINN model predicted the transient temperature field with sparse data while minimizing loss functions, and the predictions were compared to the reference values. The results show stable convergence and high prediction accuracy of the reduced-dimension PINN model compared to the three-dimensional transient fin model. Furthermore, this study shows the effectiveness of PINN under different values of the convective heat transfer coefficient (h) and noisy environments.

This study demonstrates the high accuracy of the PINN for transient fin problems by comparing the results with data from virtual sensors obtained from simulations of an industrial numerical solver. To do this, this study presents a novel approach of reducing the dimensions and developing training algorithms. The proposed approach demonstrates that PINNs can provide an accurate solution without conventional mesh discretization for transient fin problems, serve as a competitive alternative to classical numerical solvers, and can be a computationally cost-effective solution for complex geometries.

Keywords: Physics-informed neural network (PINN); Transient heat transfer; Straight fin; Thermal analysis; Heat conduction; Convective heat transfer; Machine learning; Sparse data

1 Introduction

Heat transfer is a major concern in engineering fields. Effective thermal management is essential in systems consisting of extended surfaces (fins), which are used to enhance heat dissipation. Overheating can cause material properties to degrade over time, reducing system performance and potentially leading to component failure. Therefore, accurate approximation of the temperature field of a system is considered an important step in designing reliable and efficient engineering systems.

Modern engineering system design is highly dependent on numerical solvers or simulation software. Most of these tools use the finite element method (FEM) to solve such tasks. Although FEM provides an accurate solution, it is often computationally intensive and time consuming. In addition, conventional numerical solvers require complete input data including the geometric model, mesh, material properties, and boundary conditions. Mesh generation is computationally expensive and time consuming for complex geometry.

To overcome these limitations, researchers have developed a solution named Physics-Informed Neural Network (PINN). The modern form of PINNs was introduced by Raissi et al. [1] as a tool or framework for solving forward and inverse problems containing nonlinear partial differential equations (PDEs).

Unlike conventional data-driven numerical solutions, PINNs incorporate physical laws into the loss functions, use automatic differentiation, and require little or no labeled data. Furthermore, PINNs eliminate the need for traditional mesh-based discretization, thus reducing computational cost for problems involving complex geometry. For these advantages, PINNs have gained significant attention in recent years and have been successfully applied to a wide range of engineering problems, including heat transfer, fluid mechanics, and material engineering [2, 3, 4].

The ability of PINNs to integrate physics with observable data has made them an attractive alternative to conventional finite element method (FEM) solvers for many applications. Therefore, numerous improvements and variants of the PINN framework have been proposed and adapted due to improved convergence, computational efficiency, and prediction accuracy [5, 6, 7].

Recent studies have demonstrated the potential of PINNs as an alternative computational framework to solve heat transfer problems.

Garcia-Ferreira and Aguirre [8] investigated the application of PINNs for predicting two-dimensional steady-state temperature fields and estimating unknown convection coefficients from sparse temperature data. However, the study was limited to a simple steady-state two-dimensional domain and did not consider practical engineering applications such as extended surfaces.

Also, Oliveira et al. [9] applied PINNs to solve inverse heat transfer problems. Nevertheless, their work focused mainly on inverse heat conduction problems and did not evaluate the performance of the forward heat transfer analysis or compare their effectiveness with the conventional numerical solver.

Furthermore, Li et al. [10] proposed an improved PINN framework. However, the increased complexity of the model may introduce additional computational and implementation challenges. Moreover, their approach primarily focused on heat conduction without considering the combined effects of conduction and convection, which is essential in fin heat transfer analysis.

In general, these studies confirm the capability of PINNs for heat transfer analysis. However, most previous studies were limited to steady-state problems or focused largely on the conduction term.

Existing research remains limited regarding the applicability and reliability of PINNs for extended surface heat transfer problems, particularly for fin temperature prediction under realistic transient engineering conditions.

A fast and accurate predictive model can support thermal design optimization for the iterative design process. Moreover, a lightweight PINN model has the potential to be integrated into the control logic of embedded devices for real-time temperature prediction. Such integration can enable adaptive thermal management rather than predefined thermal control schemes, thereby improving energy efficiency in engineering devices.

Therefore, further investigation is required to evaluate the accuracy, efficiency, and applicability of PINNs for fin-based systems compared to conventional numerical methods.

This study developed a PINN framework with a novel approach for predicting the transient temperature field of a three-dimensional straight fin with realistic material and geometric properties.

To construct a lightweight and computationally efficient PINN framework, the three-dimensional fin problem is reduced to a one-dimensional formulation through appropriate mathematical modeling. Despite the dimensional reduction, this study demonstrates that the one-dimensional PINN can reproduce a full temperature field of a three-dimensional fin. The reference solutions are obtained using a conventional numerical solver with three-dimensional fin geometry for validation.

Throughout the investigation, the applicability, accuracy, and reliability of PINN for extended surface heat transfer problems are evaluated, addressing the research gap in existing PINN-based thermal studies.

The performance of the proposed framework is evaluated by comparing its predictions with conventional numerical solutions, demonstrating the capability of PINNs to solve transient fin temperature prediction problems. Furthermore, the robustness of the proposed PINN framework is evaluated under noisy data conditions to assess its applicability for practical engineering applications and real-world experimental environments.

2 Mathematical Formulation

2.1 Physical Model Description

For this study, a three-dimensional straight vertical fin subjected to transient heat transfer under natural air convection and without internal heat generation is considered. The fin has homogeneous material properties and uniform cross-section. The fin is attached to a uniformly heated base and exposed to ambient air at room temperature. Heat conduction occurs from the base along the fin length and convection occurs from all exposed surfaces to the surrounding fluid (air). Heat transferred by radiation is assumed negligible and excluded from the present study. Initially, the fin is assumed to be at a uniform temperature equal to the ambient air temperature. The fin base is subjected to a constant uniform heat flux throughout the transient analysis.

2.2 Governing Equation for Three-Dimensional Transient Fin

The temperature distribution in the three-dimensional transient fin is governed by the following equation:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \alpha \left[\frac{\partial^2 T(x,y,z,t)}{\partial x^2} + \frac{\partial^2 T(x,y,z,t)}{\partial y^2} + \frac{\partial^2 T(x,y,z,t)}{\partial z^2} \right] \tag{1}$$

where, $\alpha = \frac{k}{\rho c_p}$.
For this study, initial condition is $T(x,y,z,0) = T_i$, where $t = 0$ and T_i is uniform initial temperature.
Boundary condition at base defined as

$$-k\nabla T(x,y,z,t) \cdot \hat{n} = q''$$

where, q'' is constant heat flux.
For all exposed surfaces of the fin, the condition is:

$$-k\nabla T(x, y, z, t) \cdot \hat{n} = h\left[T(x, y, z, t) - T_\infty\right]$$

And the temperature gradient is defined as:

$$\nabla T(x, y, z, t) = \frac{\partial T}{\partial x}\hat{i} + \frac{\partial T}{\partial y}\hat{j} + \frac{\partial T}{\partial z}\hat{k}$$

In this study, these equations are valid within the domain $\Omega \subset \mathbb{R}^3$.

The reference model has a three-dimensional geometry, while the PINN formulation assumes that the dominant heat transfer changes occur along one spatial direction. This assumption is justified for this study due to the geometry of the reference model. Consequently, a reduced one-dimensional model is adopted for the development of the PINN.

2.3 Applicability of One-Dimensional Equation for Physics-Informed Neural Network

The fin considered in this study has a uniform cross-section in the y-axis and z-axis, while its length along the x-axis. The length of the fin is significantly greater than the thickness of fin. These geometric characteristics results in a high slenderness ratio. For such thin fin, heat conduction across the cross-section (y and z directions) occurs much more rapidly than along the fin length.

Therefore, the temperature within each cross-sectional slice quickly become uniform. As a result, the temperature variation is assumed to occur only along x-axis. This allows each cross-section to be represented by a single temperature value T(x, t).
Thus, $\frac{\partial T}{\partial y} \approx 0, \quad \frac{\partial T}{\partial z} \approx 0$ and $T(x, y, z, t) \approx T(x, t)$

Convection is transformed from a three-dimensional surface boundary condition into a one-dimension distributed sink term by representing the surface heat loss as an equivalent volumetric heat sink. This approach mathematically distributes the effect of surface convection throughout the volume using lumped volumetric representation of convective heat loss. Thus, reducing the complexity of surface convection calculation in PINN to multi-dimension to single-dimension.

This method holds true when fin Biot number is much less than 0.1, indicating the temperature gradients within the fin cross-section are negligible because the internal thermal conductivity is much stronger than surface convection. Heat is conducted much faster than it is removed from the surface, causing cross-section to remain nearly isothermal. In this condition, the temperature variation mainly occurs along the length of the fin. The surface and internal points of a single cross-section both have almost identical temperature. Thus, this single temperature value in a cross-section can be used in the total surface heat loss function of a cross-section. Then this heat loss can be assigned to the corresponding volume and represented as a volumetric sink term. Under this Biot number condition, it is justified to represent multi-dimensional convection to one-dimensional distributed sink term.

The fin considered for this study has a Biot number of $Bi \approx 9.8 \times 10^{-5}$ and slenderness ratio of 10.

Although the Biot number varies with the convective heat transfer coefficient (h), the Biot number reamins below 0.1 for all cases considered in this study.

These values support the use of one-dimensional equation for PINN framework with other assumptions, and the results are compared against the reference solution obtained from a three-dimensional model.

2.4 Governing Equation for One-Dimensional Transient Fin

Convective heat loss occurs on all exposed surfaces of the fin in the three-dimensional model. For the reduced-order PINN, the effect is represented as a distributed volumetric sink term in a one-dimensional fin governing equation which accounts for the side surface (lateral surface) heat loss. Additionally, a tip boundary condition is added for convective heat loss from the fin tip. These equations enable the reduction of the original three-dimensional heat transfer problem to a one-dimensional problem. The accuracy of the reduced model and errors associated with dimensional reduction are evaluated against the three-dimensional reference model . The transient temperature distribution in the one-dimensional fin is governed by the following equation:

$$\frac{\partial T(x,t)}{\partial t} = \alpha \frac{\partial^2 T(x,t)}{\partial x^2} - \frac{hP}{\rho c_p A} \left(T(x,t) - T_\infty\right) \tag{2}$$

which is valid for $x \in (0, L), \quad t > 0$.
The initial condition is defined as:

$$T(x,0) = T_i \quad \text{when} \quad 0 \leq x \leq L$$

The boundary Conditions are given as follows:
For the fin base,

$$-k\frac{\partial T(0,t)}{\partial x} = q'' \quad \text{when} \quad t > 0$$

For the convective fin tip,

$$-k\frac{\partial T}{\partial x}(L,t) = h\left(T(L,t) - T_\infty\right) \quad \text{when} \quad t > 0$$

The distributed volumetric sink term is defined as:

$$-\beta\left(T(x,t) - T_\infty\right), \qquad \text{where } \beta = \frac{hP}{\rho c_p A}$$

The one-dimensional problem is defined over the following space and time domain: $x \in [0, L], \quad t \in [0, t_f]$

2.5 Error Metrics

The accuracy of the reconstructed temperature field is evaluated using the Root Mean Square Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_i^{\text{pred}} - T_i^{\text{true}}\right)^2}$$

where, T_i^{pred} is the predicted temperature, T_i^{true} is the reference temperature, and N is the total number of measurements.

2.6 Nomenclature

t_f = Final simulation time
L = Characteristic length of the domain in x-direction
A_t = Area of the tip surface
$T(x, y, z, t)$ = Temperature field
x, y, z = Spatial coordinate along the fin body
t = Time
α = Thermal diffusivity
h = Convective heat transfer coefficient
P = Perimeter of the fin cross-section
$A_c = A$ = Cross-sectional area
ρ = Density
$c_p = c$ = Specific heat capacity
T_∞ = Ambient temperature
n = Outward normal vector to the surface
∇T = Gradient of temperature

3 Numerical Simulation

3.1 Numerical Model Setup

A three-dimensional transient thermal simulation was performed using the Transient Thermal module in ANSYS Workbench 2026 R1 to investigate the heat transfer characteristics of the studied fin. The numerical model is developed using the same geometry, material properties, and boundary conditions as those employed in the PINN model to ensure a valid and consistent comparison. For consistency with the PINN simulation, heat transfer by radiation is not considered in numerical analysis. The governing assumptions, equations, material properties, initial conditions, and boundary conditions described in the mathematical formulation were directly implemented in the numerical solver.

The fin was modeled using a custom-defined aluminum with homogeneous properties which represents engineering-grade aluminum. The material properties, including density, thermal conductivity, specific heat capacity, were specified based on realistic engineering values rather than the default pure aluminum in the solver library.

The computational domain was discretized using a three-dimensional mesh. Four temperature sensor probes were positioned along the fin length at 20%, 40%, 60%, 80% of the total fin length. At each location, the sensors were positioned at the mid-plane of y-z cross-section. These sensors were used to extract point-wise temperature data from the numerical solver, which were used as reference data.

The transient analysis was performed over a specified time interval using the program-controlled time stepping option in the solver. This feature automatically adjusts the time step size to ensure numerical convergence and stability. Time integration was enabled with an automatically selected initial time step of 1 s, and the final simulation time was set to 100 s.

3.2 Physical and Material Properties

Table 1: Dimensions of the fin.

Fin Parameter	Value (mm)	Coordinate Direction
Length	100	X-axis
Width	10	Y-axis
Height	10	Z-axis

Table 2: Properties of the studied aluminum sample.

Material Property	Symbol	Value	Unit
Density	ρ	2700	kg m^{-3}
Thermal conductivity	k	205	W m^{-1} K^{-1}
Specific heat capacity	c_p	900	J kg^{-1} K^{-1}

3.3 Transient Solver Tuning

h is varied in each simulation. The values are 4, 8, 12, 24, and 48.

Table 3: Simulation conditions.

Simulation Condition	Symbol	Value	Unit
Total heat input	Q	2	W
Base heat flux	q''	20000	W m^{-2}
Initial fin temperature	T_0	25	°C
Ambient (bulk) temperature	T_∞	25	°C
Convective coefficient	h	4, 8, 12, 24, 48	W m^{-2} K^{-1}
Total simulation time	t_f	100	s
Time stepping	–	Program Controlled	–

3.4 Mesh Independence Study

A mesh independence study was performed using an identical simulation setup with a constant convection coefficient of h = 8 for different mesh sizes: 1 mm, 2 mm, 4 mm, and 8 mm. The transient average temperature of the fin was monitored at different time instants throughout the simulation. The 1 mm mesh was considered the finest mesh and was selected as the reference mesh. At each time step, the average fin temperature value obtained from the reference mesh was defined as T_{ref} and used for comparison.

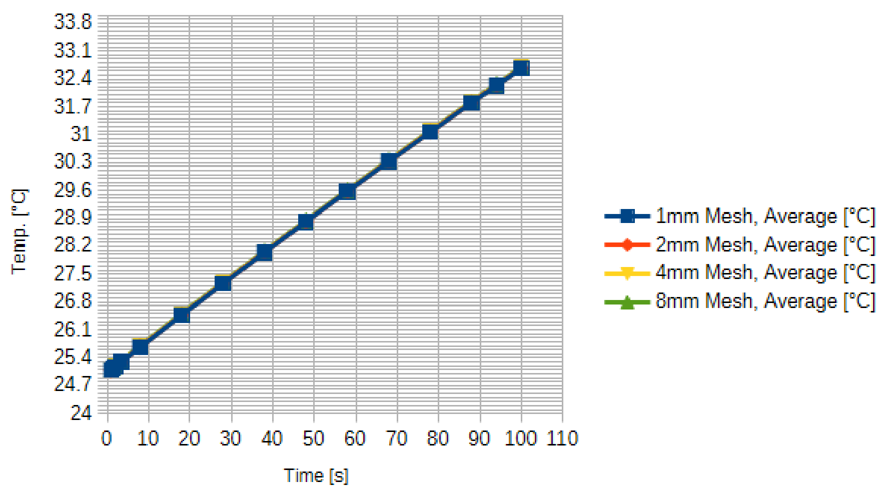

Figure 1: Temperature variation with time for different mesh sizes.

Table 4: Mesh data and error comparison.

Mesh Size (mm)	Nodes	Elements	T_{avg} at $t = 100$ s	Mean Relative Error	Max. Relative Error
1	46541	10000	32.666 °C	–	–
2	6696	1250	32.672 °C	0.0180	0.0234
4	1440	225	32.683 °C	0.0539	0.0642
8	164	13	32.701 °C	0.1145	0.1364

The errors are calculated at each time step using the following expressions: Relative Error,

$$E_i(t_j) = \frac{|T_i(t_j) - T_{\text{ref}}(t_j)|}{T_{\text{ref}}(t_j)} \times 100$$

Mean Relative Error,

$$E_{\text{mean},i} = \frac{1}{N} \sum_{j=1}^{N} E_i(t_j)$$

Maximum Relative Error,

$$E_{\max,i} = \max\left(E_i(t_j)\right)$$

Where, T_i represents the average temperature of fin obtained from mesh size i (2 mm, 4 mm and 8 mm mesh) at each time step j, and T_{ref} is the average temperature of fin obtained from the 1 mm reference mesh.

The results demonstrate excellent numerical convergence among different mesh sizes. The 2 mm mesh shows a mean relative error of 0.018% and a maximum relative error value of 0.023% compared with the reference mesh. Even for the coarsest 8 mm mesh, the maximum relative error remained below 0.14%. These results indicate that the numerical solution becomes mesh independent for mesh sizes of 2 mm or finer, and further refinement does not provide any significant improvement in accuracy. Consequently, the 1 mm mesh (finest) was selected as reference mesh size for the rest of the study.

3.5 Numerical Data Extraction

The transient temperature results were extracted from the numerical solver solution at predefined monitoring sensor locations. The obtained temperature data were used as reference solutions for evaluating the PINN framework. The mesh independence study was performed separately based on the average temperature of the entire fin body, as it represents global thermal behavior rather than the sensor data, which localized the thermal behavior. This approach ensures that the selected mesh provides sufficient numerical accuracy for the entire fin and is not limited to specific sensor locations.

4 PINN Implementation

4.1 PINN Approximation

The objective of this study is to reconstruct the temperature field $T(x,t)$ by employing a Physics-Informed Neural Network (PINN). The heat transfer coefficient (h) is assumed to be known, and the neural network is trained using sparse temperature measurements together with the governing heat equation. The PINN approximates the temperature field $T(x,t)$ by a neural network function T_θ which is parameterized by the trainable parameter θ such that, $T_\theta(x,t) \approx T(x,t)$. θ represents all the weights and biases in the neural network.

4.2 Network Architecture

Table 5: Neural Network Architecture

Parameter	**Value**
Number of input features	2
Number of hidden layers	5 layers
Neurons per hidden layer	64 neurons
Neurons in output layer	1 neuron
Number of total layers (including input and output)	7 layers
Hidden layer activation function	tanh
Output layer activation function	Linear
Stage 1 optimizer	Adam optimizer
Stage 2 optimizer	L-BFGS optimizer

Table 6: Optimizer Settings

Parameter	**Value**
Number of epochs for Adam optimizer	2000
Learning rate (Adam optimizer)	1×10^{-3}
Maximum number of iterations (L-BFGS)	500
Learning rate (L-BFGS)	1.0
History size (L-BFGS)	50
Gradient tolerance	1×10^{-9}
Change tolerance	1×10^{-9}
Line search algorithm	Strong Wolfe

Here,

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

4.3 Governing Equation, Constraint and Residual

The governing equation, initial condition, and boundary conditions of the one-dimensional fin described in Section 2.4 are incorporated into the PINN framework as constraints. These constraints guide the neural network during the training process by enforcing heat transfer physics. These equations and conditions are transformed into corresponding residual functions, and the loss functions are defined based on the mean squared error (MSE) of these residuals.

Physics residual defined as:

$$R_f(x,t) = \frac{\partial T_\theta(x,t)}{\partial t} - \alpha \frac{\partial^2 T_\theta(x,t)}{\partial x^2} + \frac{hP}{\rho c_p A}\left(T_\theta(x,t) - T_\infty\right)$$

Initial condition residual defined as:

$$R_{\mathrm{IC}}(x,0) = T_\theta(x,0) - T_i$$

Boundary condition residual defined as:

$$R_{base}(t) = -k\frac{\partial T_\theta(0,t)}{\partial x} - q''$$

and

$$R_{tip}(t) = -k\frac{\partial T_\theta(L,t)}{\partial x} - h\left(T_\theta(L,t) - T_\infty\right)$$

Data residual defined as:

$$R_{\mathrm{data}}(x,t) = T_\theta(x,t) - T_{\mathrm{sensor}}(x,t)$$

Here, T_i is the initial condition and $T_{\mathrm{sensor}}(x,t)$ is extracted temperature data from sensor probes in solver.

The corresponding loss functions are formulated as:
Physics loss,

$$L_f = \frac{1}{N_f}\sum_{i=1}^{N_f} \left|R_f(x_i,t_i)\right|^2$$

Initial condition loss,

$$L_{\mathrm{IC}} = \frac{1}{N_{\mathrm{IC}}}\sum_{i=1}^{N_{\mathrm{IC}}} \left|R_{\mathrm{IC}}(x_i,0)\right|^2$$

Boundary loss,

$$L_{\mathrm{BC}} = \frac{1}{N_{\mathrm{base}}}\sum_{i=1}^{N_{\mathrm{base}}} \left|R_{\mathrm{base}}(t_i)\right|^2 + \frac{1}{N_{\mathrm{tip}}}\sum_{i=1}^{N_{\mathrm{tip}}} \left|R_{\mathrm{tip}}(t_i)\right|^2$$

Data loss,

$$L_{\text{data}} = \frac{1}{N_d} \sum_{i=1}^{N_d} |R_{\text{data}}(x_i, t_i)|^2$$

The total loss function is defined as,

$$L_{\text{total}}(\theta) = \lambda_f L_f + \lambda_b L_{\text{BC}} + \lambda_i L_{\text{IC}} + \lambda_d L_{\text{data}}$$

Where, $N_f, N_{\text{IC}}, N_{\text{Base}}, N_{\text{Tip}}, N_d$ represent the number of randomly selected points used for evaluating the corresponding residuals. These point sets are independently sampled.
$\lambda_f, \lambda_b, \lambda_i, \lambda_d$ are weighting coefficients for physics, boundary condition, initial condition and data losses, respectively.
The PINN training process finds the optimal neural network by minimizing the total loss function, which can be expressed as:

$$\min_{\theta} L(\theta)$$

Here, $L(\theta)$ is the total loss function dependent on the trainable parameter θ.

4.4 Sampling Strategy for PINN Training Points

Table 7: Sampling Strategy for Training Points

Constraint Type	**Notation**	**Domain Definition**	**Sampling Strategy**	**Number of Points**
Physics	N_f	$x \in [0, L]$ $t \in [0, t_f]$	Random uniform sampling of space and time. Independent.	5000
Boundary Conditions	N_{base} N_{tip}	$x = 0$ base $x = L$ tip $t \in [0, t_f]$	Random uniform time sampling at base and tip.	1000 base 1000 tip
Initial Condition	N_{IC}	$t = 0$ $x \in [0, L]$	Random uniform spatial sampling at initial time.	1000
Data	N_d	$x = \{0.2L, 0.4L, 0.6L, 0.8L\}$ $t \in [0, t_f]$ $h = \{4, 8, 12, 24, 48\}$	Random time snapshots at fixed sensor locations for each h.	24 per h

For each convection coefficient h case, 24 temperature measurements were used as data points. The data points were randomly sampled from four virtual sensors, with six measurements selected from each sensor at different time snapshots. The sampling was performed independently for each sensor and each h case, resulting in different time snapshots being selected without any predefined pattern to reduce sampling bias during PINN training.

4.5 Loss Weights

To balance the contribution of different loss components during the PINN training process and prevent any single loss term from dominating the optimization, weighting coefficients are assigned to each loss component. The selected values for loss weights are:

Table 8: Values of Loss Weights

Loss Weights	**Physics Loss**	**Boundary Loss**	**IC Loss**	**Data Loss**
Value	1	1	1	10

The data loss weight was increased to emphasize agreement with sparse data which contain direct information about temperature field.

4.6 Input and Output Normalization

The input variables of the neural network have different physical scales. Neural networks usually achieve better training when the input variables have similar numerical range. For this purpose, input normalization is applied in this study. After normalizing the inputs, variables transformed as:

$$x^* = \frac{x}{L}, \qquad t^* = \frac{t}{t_{\max}}$$

These normalized variables are provided as inputs to the neural network. The neural network predict normalized temperature output. For further study, the normalized output is converted back to the original temperature scale using:

$$T_{\text{denormalized}} = T_{\text{normalized}} \times T_{\text{scale}} + T_{\text{ref}}$$

The T_{ref} is the reference temperature, selected as the ambient temperature and the T_{scale} is temperature scaling factor.

Temperature scaling factor selected value is 40 °C, which is the slightly higher than maximum observable temperature in sparse measurement data. Both these value can be selected in expected range, since they only affect numerical scaling and simplify for the normalization process. During training, the sparse temperature data are retained in their original scale. The normalized neural network output is first transformed back to original scaling using the inverse normalization relation above. The denormalized prediction is then used for both the calculation and comparison with the sparse measurements to evaluates the loss. In this study, this novel approach achieves an excellent convergence and high accuracy in results.

4.7 Automatic Differentiation

The derivatives required in the governing equation are obtained using automatic differentiation(AD) of the neural network output. A common approach in PINN studies is to perform automatic differentiation on the normalized neural network output and subsequently transform the obtained derivatives back to the original physical scale.

Initially, this study adopted this approach; however, it was found that the rescaling of derivatives introduced significant errors in final results. To address this issue, a novel approach was developed and adapted in this study and PINN framework.

The newly developed algorithm denormalize the predicted temperature values by transforming them back to their original physical scale. After the rescaling, the automatic differentiation calculate derivatives over the original physical scale which resulted in a significant reduction in errors.

4.8 Training and Optimization

The PINN was trained using both the governing equation and sparse data obtained from numerical solver outputs at selected sensor locations along the fin length. The sensor probes were uniformly placed along the fin length at 0.2L, 0.4L, 0.6L, and 0.8L to reduce bias in the extracted data.

During each session, a fixed number of randomly selected samples from the sparse dataset were provided to the PINN, and the network was trained for different convective heat transfer coefficient (h) values. This approach represents a practical engineering scenario where limited sensor feedback is available. Since many thermal management modules utilize sensor-based feedback mechanisms, integrating sparse measurement data with the governing equation can improve the applicability of PINN models in practical applications.

A two-stage optimization strategy was adopted to improve the convergence and accuracy of the PINN model. In the first stage, the Adam optimizer was employed for 2000 epochs with the learning rate and other optimizer settings provided in Table 6. In the second stage, the L-BFGS optimizer was applied to further refine the network parameters obtained from the Adam optimization stage. The corresponding L-BFGS settings are also mentioned in Table 6. The optimization process continued until the stopping criteria were satisfied or the maximum number of iterations was reached.

4.9 Software and Hardware

The PINN model was developed using the Python programming language with the PyTorch framework, along with NumPy and Pandas libraries. The neural network architecture was manually developed using PyTorch modules, allowing fine-tuning of the network layers, activation functions, loss formulation, and optimization process. NumPy was used for numerical data processing and array manipulation, while Pandas was used for sparse data handling and data import. The PINN training and evaluation were performed using a CPU-based computational environment.

4.10 Convergence Criteria

The convergence of the PINN model was evaluated by monitoring the implemented total loss function during the optimization process. During the Adam optimization stage, the loss value was monitored throughout the epochs to evaluate the reduction of loss and the stability of the optimization process. The network parameters were updated through backpropagation until the predefined number of epochs was completed.

In the L-BFGS optimization stage, the loss evolution was also monitored during the process. The optimization terminated when the gradient tolerance or change tolerance criteria were satisfied, or when the maximum number of iterations was reached.

4.11 Evaluation

The performance of the trained PINN model was evaluated using unseen test data that were not provided during the training process. The training and test datasets were obtained from the numerical solver. The prediction accuracy was quantified using the root mean square error (RMSE), which measures the error magnitude between the PINN predictions and the reference solutions. The temperature field is represented in degrees Celsius ($^\circ$C) in both the numerical solver output and the PINN predictions. Since the governing equation depends only on temperature difference, and temperature increments satisfy 1 $^\circ$C increment equivalent to 1 K increment, no conversion to Kelvin is required. Therefore, all computations are performed directly using Celsius scale.

The RMSE was calculated over all unseen test points corresponding to different spatial locations and time instants for each convective coefficient case.

4.12 Algorithm Pseudocode

Algorithm 1 PINN Algorithm Pseudocode

1: **Input:**
    Governing equation
    Boundary Conditions
    Initial Conditions
    Sparse Data
2: Initialize the PINN model with the selected network architecture.
3: Import the sparse temperature data from the dataset file.
4: Generate all collocation points.
5: Normalize the input variables (space and time).
6: Feed the normalized inputs into the PINN model.
7: Predict temperature field using PINN.
8: Denormalize the predicted temperatures with the appropriate scaling factors.
9: Differentiate the denormalized temperature.
10: Repeat steps 5-9 independently for each residual function.
11: Calculate the total loss from the residual functions.
12: Perform two-stage optimization.
13: Optimize the total loss using backpropagation.
14: Check the convergence criteria.
15: If the convergence criteria are satisfied, stop the optimization process.
16: Otherwise, continue the optimization process until convergence is achieved.
17: **Output:**
    Trained PINN model
18: For testing the PINN model, feed unseen testing data into the trained PINN.
19: Calculate the RMSE between the predicted solution and the reference data.
20: Repeat the complete algorithm from Step 1 independently for each convective coefficient (h) value, using a newly initialized PINN without any previous training information.
21: **Output:**
    Predicted solution for each h value

Other fixed parameters required to satisfy the governing equation and remain constant for all h cases are not included in the pseudocode. The value of h is assigned in code separately for each individual case.

4.13 Noise Robustness Test

To evaluate the robustness of the proposed Physics-Informed Neural Network (PINN), Gaussian noise was added to the temperature values of the training data. The noisy temperature field was generated as:

$$T_{\text{noise}} = T_{\text{true}} + \delta\sigma_{\text{true}}\epsilon$$

where δ is the noise level, σ_{true} is the global standard deviation of the true temperature data, and ϵ is a random variable sampled from a standard normal distribution with zero mean and unit variance.
Noise levels of 1%, 3% and 5% were considered. The same set of random values of ϵ was used for all noise levels to ensure a fair comparison. The testing data remained noise-free to evaluate the ability of the PINN to reconstruct the true solution from noisy training data.

5 Results and Discussion

Table 9: Prediction Accuracy of the PINN Model for Different h Values

Case	**Convective coefficient, h (W m^{-2} K^{-1})**	**RMSE (°C)**
A	4	0.3674800875514497
B	8	0.4064575181879708
C	12	0.31703886718915353
D	24	0.8862680961379742
E	48	0.40470020132672263

Table 10: Effect of Training Data Noise on Model Accuracy

Convective Coefficient, h (W m^{-2} K^{-1})	**Case**	**Noise Level**	**RMSE (°C)**
4	Mild	1%	0.3443709314000222
4	Moderate	3%	0.40362959099263596
4	Strong	5%	0.35376694335295683

The RMSE was calculated using the temperature difference between the PINN prediction and the three-dimensional reference solution; therefore, the obtained RMSE values represent absolute temperature errors in °C.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_{\text{PINN},i} - T_{\text{ref},i})^2}$$

Where, T_{PINN} is the predicted temperature (°C) and T_{ref} is the reference temperature (°C), and N is the number of temperature data points.

The performance of the proposed reduced-order PINN framework was evaluated under different convective heat transfer coefficients (h) and training data noise levels. Table 9 presents the prediction accuracy of the PINN model under different convective heat transfer coefficients. The model achieved relatively low RMSE values for all investigated cases, demonstrating good agreement between the temperature fields reconstructed by the reduced-order PINN model and the three-dimensional reference data. The lowest prediction error was obtained for Case C (h=12) with an RMSE of 0.3170, whereas the highest error was obtained for Case D (h=24), with an RMSE of 0.8863. The variation in RMSE does not indicate a direct relationship between the convective heat transfer coefficient and prediction error.

For instance, both Case A (h=4) and Case E (h=48) produced lower RMSE values than Case D. This behavior suggests that the increased error in Case D is mainly associated with the optimization process rather than the effect of the convective coefficient itself.

During the training of Case D (h=24), the L-BFGS optimizer reduced the loss function from 19,452.34 at the beginning of Stage 2 optimization to approximately 13.97 at the final iterations, indicating effective optimization progress. However, the optimization was terminated when one of the predefined L-BFGS stopping criteria was satisfied. Consequently, the training process resulted in a converged solution with a higher RMSE compared with the other cases. Since PINN accuracy is highly dependent on the final optimization process, further improvement of this case may be possible through appropriate adjustment of L-BFGS parameters.

Table 10 presents the effect of noise on PINN prediction accuracy for the convective coefficient h=4. The model maintained stable prediction performance even when noise was introduced into the training data. The RMSE values were 0.3444, 0.4036, and 0.3538 for noise levels of 1%, 3%, and 5%, respectively.

The limited variation in prediction error demonstrates the robustness of the PINN approach in reconstructing the transient temperature field under uncertain measurement conditions.

Overall, the proposed reduced-order PINN framework demonstrates the capability to reconstruct the transient temperature field of a three-dimensional fin using a simplified one-dimensional formulation. The three-dimensional reference data were generated from the complete three-dimensional fin model, while the PINN was trained using the reduced one-dimensional governing equation with appropriate assumptions and boundary conditions.

The good agreement between the reconstructed results and the three-dimensional reference data confirms that the reduced-order approach can effectively capture the transient thermal behavior of the original system. Furthermore, the robustness under noisy conditions highlights its potential applicability in thermal systems where complete high-dimensional measurements are difficult to obtain. This study also demonstrates the applicability of PINNs as an efficient approach for transient thermal analysis using reduced-order information.

6 Conclusion

This study developed a reduced-order physics-informed neural network (PINN) framework for reconstructing the transient temperature field of a three-dimensional fin using a simplified one-dimensional physics formulation. The proposed approach successfully demonstrated that high-dimensional thermal behavior can be captured using reduced-order information while maintaining good agreement with three-dimensional reference solutions. Furthermore, the reduced-order formulation decreases the computational complexity compared with directly solving full three-dimensional PINN models or higher-dimensional numerical approaches.

The model achieved low prediction errors, with RMSE values below 1 °C for all investigated cases, and maintained stable performance under noisy training data. The results confirm that the proposed reduced-order PINN approach can effectively reconstruct transient thermal fields while reducing the complexity of transient thermal analysis and preserving prediction accuracy.

The main contribution of this work is the demonstration of dimensional reduction combined with PINNs for efficient thermal field reconstruction. Future work will focus on improving optimization strategies, extending the framework to more complex geometries and validating the approach using experimental temperature measurements.

7 Acknowledgments

The author has no acknowledgments to declare.

8 Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

9 Conflict of Interest Statement

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

10 Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

11 CRediT authorship contribution statement

Md. Muminul Islam: Conceptualization, Methodology, Software, Investigation, Data curation, Formal analysis, Visualization, Validation, Writing – original draft, Writing – review & editing.

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*, 378:686–707, 2019.

[2] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 04 2021.

[3] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: a review. *Acta Mechanica Sinica*, 37(12):1727–1738, 12/01 2021.

[4] Alkmini Michaloglou, Ioannis Papadimitriou, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. Physics-informed neural networks in materials modeling and design: A review. *Archives of Computational Methods in Engineering*, 33(4):5223–5260, 05/01 2026.

[5] Kuang Luo, Jingshang Zhao, Yingping Wang, Jiayao Li, Junjie Wen, Jiong Liang, Henry Soekmadji, and Shaolin Liao. Physics-informed neural networks for pde problems: a comprehensive review. *Artificial Intelligence Review*, 58(10):323, 07/24 2025.

[6] Zhuang Zhao, Ye Wang, Weijian Zhang, Zhenggang Ba, and Lin Sun. Physics-informed neural networks in heat transfer-dominated multiphysics systems: A comprehensive review. *Engineering Applications of Artificial Intelligence*, 157:111098, 2025.

[7] Cuomo, Salvatore and Di Cola, Vincenzo Schiano and Giampaolo, Fabio and Rozza, Gianluigi and Raissi, Maziar and Piccialli, Francesco. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing*, 92(3):88, 07 2022.

[8] Garcia-Ferreira, Sergio and Aguirre, Gorka. Temperature Field Prediction and Convection Coefficient Estimation from Temperature Data Using PINNs. In Wegener, Konrad and Bambach, Markus, editor, *4th International Conference on Thermal Issues in Machine Tools (ICTIMT2025)*, pages 573–587, Cham, 2026. Springer Nature Switzerland.

[9] Oliveira, Igor A. R. and Mendes, Daniel A. S. and Velho, Haroldo F. Campos and Pereira, Eduardo S. Solving backward heat transfer by physics-informed machine learning. *Computational and Applied Mathematics*, 45(10):421, 06 2026.

[10] Yuxuan Li, Weihang Liu, Jiawen Ren, Miao Cui, Zechou Zhang, and Mikhail A. Nikolaitchik. An improved physics-informed neural network

framework for solving transient nonlinear and inhomogeneous heat conduction problems. *International Journal of Heat and Mass Transfer*, 252:127515, 2025.