Precision & Performance Conference 2025, Cranfield University, UK, 17th- 20th November 2025

Physics-Informed Neural Networks for Temperature Field Reconstruction

Sergio García-Ferreira, Gorka Aguirre

IDEKO, Design and Precision Engineering Group, Elgoibar, Gipuzkoa, Spain

gaguirre@ideko.es

Abstract

Thermal deformation is a critical factor affecting the precision of machine tools, requiring accurate thermal modeling to predict temperature fields and thermal parameters. Traditional approaches, such as the Finite Element Method (FEM), require well-defined boundary conditions, which are often unknown or difficult to measure in real machining environments. This paper explores the use of Physics-informed neural networks (PINNs) as an alternative method for solving steady-state heat conduction problems in two dimensions. PINNs integrate sparse sensor data with physical laws, enabling temperature field prediction and heat transfer coefficient estimation without the need for fully specified boundary conditions. We evaluate five different PINN models, varying the balance between data-driven and physics-informed constraints. Results show that enforcing the heat equation alone yields high accuracy in temperature prediction, but accurate heat transfer coefficient estimation requires explicit enforcement of convection conditions. While PINNs successfully infer missing parameters, their sensitivity to temperature gradients can impact accuracy. Additionally, the need for retraining PINNs when conditions change limits real-time applicability, making them more suitable for offline thermal analysis rather than adaptive modelling in dynamic environments.

Temperature field reconstruction, Physics-informed neural networks, Machine Learning

1. Introduction

Thermal deformation significantly impacts machine tool accuracy due to uneven thermal expansion from temperature variations. As manufacturing demands increasingly tighter tolerances, accurate thermal modelling is crucial for high precision. However, achieving this is challenging, especially with limited sensor data, as pointwise measurements offer only partial insights into the overall temperature distribution. Traditional methods like Finite Element Method (FEM) simulations, while useful, often struggle with incomplete or hard-to-measure boundary conditions in real-world scenarios.

An alternative, Physics-informed neural networks (PINNs), combine data-driven learning with physical laws to approximate temperature fields without requiring fully defined boundary conditions. Unlike standard neural networks, PINNs incorporate governing equations (like the heat equation) directly into their loss function, ensuring physics-consistent solutions. This makes them ideal for situations with sparse sensor data, as they can infer missing information by leveraging prior physical knowledge.

The idea behind PINNs was originally introduced in the 1990s [1], but they gained significant traction after Raissi et al. popularized their modern form in 2019 [2], demonstrating their effectiveness in solving partial differential equations (PDEs) in both forward and inverse problems. Since then, PINNs have been widely applied in various scientific and engineering fields, including heat transfer.

Our research assesses PINNs' capability to predict temperature fields and estimate the heat transfer coefficient under realistic, data-scarce conditions where underlying physics may be partially known. A key aspect of our study is analysing PINNs' sensitivity to the balance between physical knowledge and available data to quantify their strengths and limitations in thermal modelling.

This paper is structured as follows: We define PINNs for both forward and inverse problems in Section 2, introduce the heat transfer problem and methodology in Section 3, present results and discussions in Section 4, and conclude with key insights and future recommendations in Section 5.

2. Physics-informed neural networks

Physics-informed neural networks (PINNs) are a class of deep learning models that integrate physical laws into the training process to approximate the solution of partial differential equations (PDEs). Unlike traditional data-driven neural networks that require big amount of data, PINNs leverage governing equations—such as conservation laws or heat transfer equations—to learn solutions in a physics-consistent manner. This makes them particularly valuable for solving inverse and forward problems where experimental data may be sparse or expensive to obtain.

PINNs are versatile tools capable of addressing both forward and inverse problems in the context of partial differential equations (PDEs). In a forward problem, PINNs act as PDE solvers for well-posed systems, meaning all initial and boundary conditions are fully defined. Here, the network learns the solution that satisfies these known conditions and the governing PDE. Alternatively, PINNs can be used inverse problems, where certain boundary conditions, parameters, or even unknown inputs within the PDE itself are undefined. Instead of explicitly specifying these missing values, PINNs incorporate observed data directly into their loss function. This allows the network to infer these unknown quantities while simultaneously solving the PDE. Essentially, PINNs combine the available data and the

underlying physics to "fill in the gaps" where information is missing.

3. Problem formulation and methodology

For an initial approach to PINNs, we consider a two-dimensional steady-state heat conduction problem with a combination of convection, fixed temperature, and adiabatic boundary conditions, as defined in Eq. (1).

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \ \forall (x, y) \in \Omega \\ k\frac{\partial u}{\partial n} = -h(u - u_\infty), \ \forall (x, y) \in \Gamma_{BCD} \\ u = u_{cte}, \ \forall (x, y) \in \Gamma_{AB} \\ \frac{\partial u}{\partial n} = 0, \ \forall (x, y) \in \Gamma_{DA} \end{cases} \quad (1)$$

No internal heat generation is considered. The problem dimensions follow the NAFEMS benchmark T4 [3], where the domain Ω is a 0.6 × 1 m rectangle. The boundary conditions are as follows: convection to an ambient temperature of 0 °C is applied on the right (BC) and top (CD) boundaries, a fixed temperature of 100 °C is imposed on the bottom boundary (AB), and a zero-heat flux (adiabatic condition) is enforced on the left boundary (DA) (Fig. 1, left). The material has a thermal conductivity of 52 W/(m°C), and the convective heat transfer coefficient on the right and top boundaries is 750 W/(m²°C). Finite Element Method (FEM) is used to obtain a reference numerical solution (ig. 1, right).

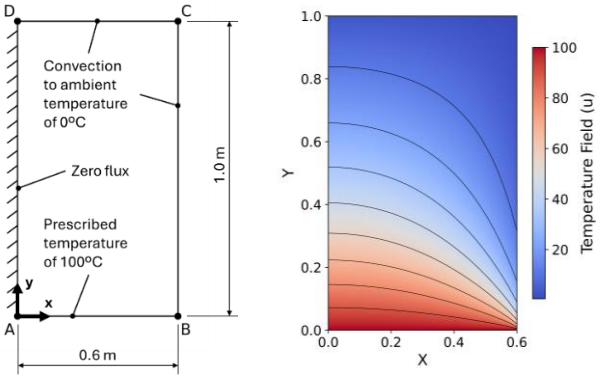

Figure 1. Left: Problem definition based on the NAFEMS benchmark T4. Right: FEM reference solution for the temperature field, with isothermal lines displayed

3.1. Methodology

Consider a solid body, such as a machine column or a spindle, where we aim to measure its temperature. Sensors are placed on its surface to collect temperature data at specific locations. However, the temperature distribution within the rest of the body remains unknown. In this study, we use this surface temperature data to predict the temperature field throughout the entire body and to estimate the heat transfer coefficient to air.

The data used in this study is obtained from simulations. Eq. (1) is solved using an in-house finite element software, and the solution is validated against the NAFEMS benchmark. FEM data is used to simulate sensor measurements (training data) and to evaluate the models after training (test data). We choose six equidistant data points per side to emulate sensors (Fig. 2).

We use PINNs to approximate the solution to the temperature field, leveraging a hybrid approach that integrates prior physical knowledge of the system with data. We analyse how this knowledge influences both the temperature field prediction and the estimation of the heat transfer coefficient h. To this end, we propose five different models based on the amount of physics and data incorporated during the neural network training. In the first three models, *h* is unknown, whereas in the last two models, *h* is known. The models range from minimal to maximal physics integration: (1) temperature data is available only at boundary data points, with no prior physics knowledge incorporated; (2) same as the previous model, but now the heat PDE is enforced over the domain; (3) same as the previous model, with the addition of convection at the right and top boundaries, though *h* remains uncertain; (4) same as the previous model, but *h* is now known; (5) the PDE and all boundary conditions are fully defined. Table 1 provides an overview of the different models and their corresponding physics and data integration levels.

Table 1 Overview of models acc. to physics and data integration.

Model	Physics integration	Data integration
1	No physics	Boundary data
2	Heat PDE	Boundary data
3	Heat PDE + convection BC (h unknown)	Boundary data
4	Heat PDE + convection BC (h known)	Left and bottom data
5	Heat PDE + all BC	No data

The PINN architecture remains the same for every model: a four-hidden-layer neural network with 20 neurons per layer and a hyperbolic tangent activation function. As there is no clear consensus on the ideal architecture for PINNs, we followed the recommendations in [4], where networks with depths of 3–6 layers and widths of 128–512 neurons are suggested. However, through empirical tuning, we observed that wider architectures did not lead to significant improvements in accuracy or convergence for our problem. Training is performed for up to 10^5 iterations, with a convergence tolerance of 10^{-6}, which serves as the stopping criterion. To evaluate the PDE, 400

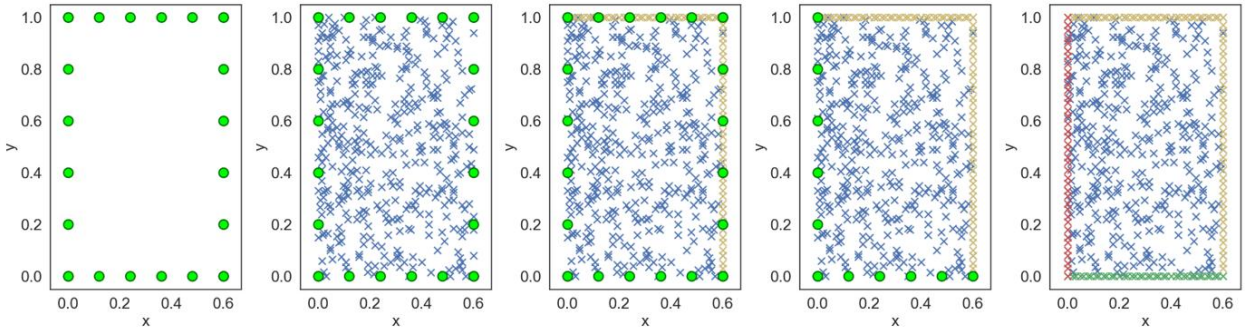

Figure 2 Data points (green circles) and residual points (coloured x markers) distribution: six data points per side, 400 residual points distributed inside the domain and 40 residual points per boundary. The images from left to right correspond to models 1 through 5.

residual points are randomly distributed within the domain, while 40 equidistant residual points per side are used to enforce boundary conditions (Fig. 2).

We use simulation data to evaluate the models at the finite element nodes, totalling 7177 test samples. Model accuracy in temperature field prediction is defined as the proportion of test samples where the prediction error is below 1% of the temperature range. Since the temperature varies from 0 to 100°C, the accuracy threshold is 1°C.

Heat transfer coefficient is evaluated in models 1 and 2 by computing the heat flux at the boundaries where convection is defined taking advantage of automatic differentiation. In model 3, it is treated as a learnable parameter, optimized alongside the neural network's trainable parameters.

4. Results and discussion

In this section, we evaluate the performance of PINNs in predicting the temperature field and estimating the heat transfer coefficient h. First, we assess the accuracy of temperature predictions across five different models, examining how incorporating physical constraints influences the solution quality. We then analyse the estimation of the heat transfer coefficient, comparing different approaches to determining *h* and identifying factors that affect its accuracy.

4.1. Temperature field prediction

We assess the temperature prediction performance of the five models outlined in the previous section. Since only boundary data is available for training in model 1, it functions as a traditional neural network rather than a PINN. The model achieves an accuracy of 23.98%, meaning that only 1 in 4 nodes within the domain is predicted with an error below 1 ºC. As expected, the most accurate predictions are concentrated near the boundaries, where the training data is provided (Fig. 3, first row). However, the model fails in the interpolation throughout the domain. Introducing prior physics in model 2, by enforcing the heat PDE, significantly improves accuracy to 98.69% (Table 2). Predictions strongly match the reference solution, and both temperature fields are barely differentiable. There is a tiny region, though, near point B (lower right corner), exhibiting a high error up to 13.18 ºC. This is most likely due to steep temperature gradients and is a challenge in PINNs that is already being addressed in literature through residual point resampling and weight adaptation.

Further refining the model, in model 3 the convection boundary conditions are incorporated while treating h as a learnable parameter. This results in an accuracy of 98.65%, the same as in the previous model. When h is known and explicitly enforced in model 4, accuracy drops to 95.92%. Finally, in model 5, where all boundary conditions are fully defined and no training data is used, the PINN operates as a pure forward solver, achieving an accuracy of 96.06% (Fig. 4).

Table 2 Accuracy in *u* temperature prediction and *h* estimation. The middle column percentages indicate what portion of the test dataset fell within a certain accuracy.

Model	Model description	*u* pred. RE < 1 %	*h* estim. accuracy
1	data	23.98 %	18.92 %
2	pde+data	**98.69 %**	51.00 %
3	pde+conv(h)+data	98.65 %	**98.45 %**
4	pde+conv+data	95.91 %	-
5	pde+conv+tcte+adiab	96.06 %	-

With these numbers in hand, it seems that adding more physics does not necessarily improve accuracy. However, even though the accuracy in model 5 is slightly lower than in models 2 and 3, the high-error region near point B has almost disappeared (Fig. 3, last row), indicating that enforcing all boundary conditions leads to a more physically consistent solution.

On the other hand, we must consider that as we added physics-based constraints to the models, we also removed data. Since neural networks learn more efficiently from data than from equation-based constraints, we expect that extending the PINN training time would eventually improve accuracy. Additionally, further tuning the loss term weights could help balance the contribution of the physics- and data-driven terms.

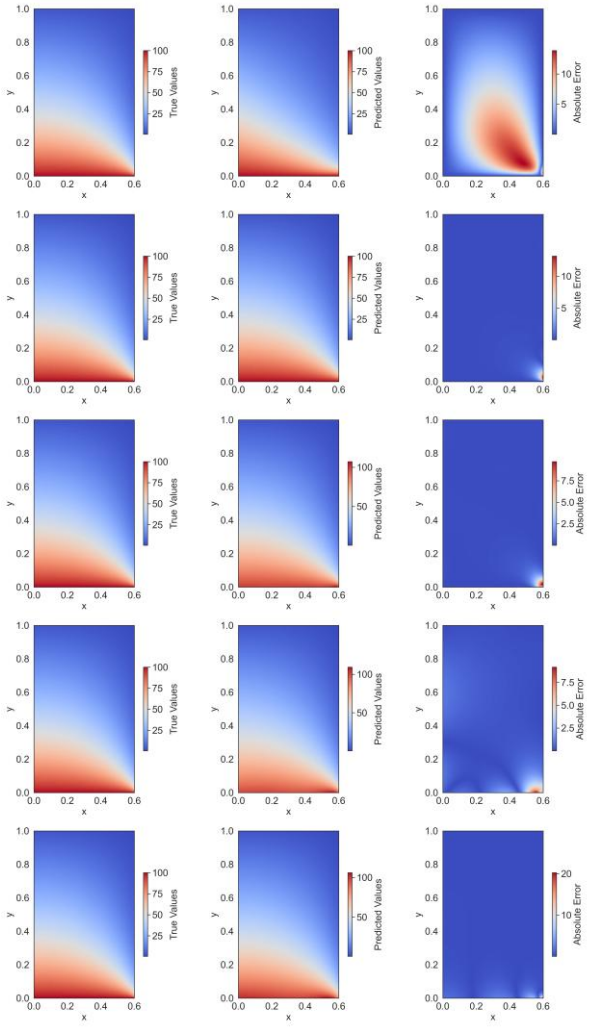

Figure 3. Temperature field predictions for each model. The first row corresponds to model 1, and the last row to model 5. The left column presents the true temperature field from the FEM simulation, the middle column shows the PINN-predicted temperature field, and the right column displays the absolute error.

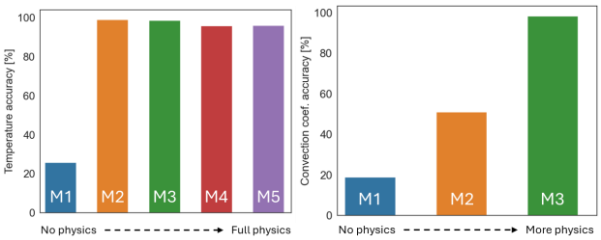

Figure 4. Temperature prediction accuracy (left) and heat transfer coefficient estimation accuracy (right). The horizontal axis represents

the degree of inclusion of physics-based constraints The vertical axis indicates temperature and heat transfer coefficient prediction accuracy.

4.1. Heat transfer coefficient estimation

We evaluate the performance of heat transfer coefficient estimation for the first three models, where *h* is treated as an unknown parameter. As described in Section 3.1, in models 1 and 2, *h* is determined *post-training*, whereas in model 3, *h* is learned *during* training. The accuracy metrics indicate that model 3 is the only one that correctly predicts *h*, achieving an estimation accuracy of 98.45%, while models 1 and 2 fail to provide accurate estimations (**¡Error! No se encuentra el origen de la referencia.**2).

The estimation of *h* depends on two key factors: the temperature field prediction, which is the output of the PINN, and the temperature gradient normal to the boundary, computed using automatic differentiation.

To further analyse this behaviour, we compute the estimated heat flux through the convective boundary and plot it against the temperature difference between the boundary and the surrounding air. Although model 2 provides the most accurate temperature field prediction, it fails to correctly estimate the heat transfer coefficient h. In model 3, the convection boundary condition is explicitly enforced, and the heat transfer coefficient h is learned during model training. As shown in **¡Error! No se encuentra el origen de la referencia.**, all data points align with the expected convection line, ensuring that the condition is satisfied across the entire domain.

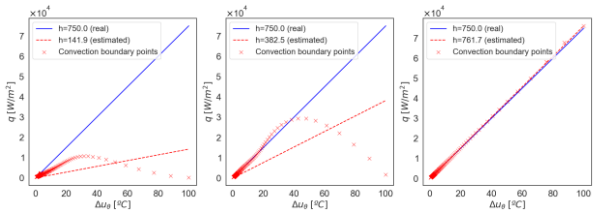

Figure 5. Heat flux through the convective boundaries as a function of the temperature difference be-tween the boundary and ambient (markers). The dashed red line represents the least-squares regression line, whose slope corresponds to the estimated heat transfer coefficient h. The solid blue line indicates the real relationship between heat flux and temperature difference based on the true h value. Left: Model 1; Middle: Model 2; Right: Model 3.

5. Conclusions

The analysis presented in this paper highlights several important findings. First, PINNs demonstrate strong capabilities in temperature field prediction when data is scarce. Results show that enforcing the heat equation alone—without requiring prior knowledge of boundary conditions—yields high accuracy in estimating the thermal state of a 2D solid surface. This suggests that PINNs are well-suited for solving inverse problems where the differential equation is known, boundary conditions are uncertain, and temperature measurements are limited. Additionally, PINNs offer a flexible modelling framework that allows for simultaneous parameter estimation and field reconstruction with no additional computational cost.

A key outcome of this study is that accurately estimating the heat transfer coefficient requires explicitly enforcing the convection boundary condition and treating the coefficient as a learnable parameter during training. This approach leads to precise estimations of the parameter, demonstrating that PINNs can successfully infer missing thermal properties when provided with suitable constraints. In contrast, post-training estimation of the heat transfer coefficient from temperature predictions alone is not reliable, as errors in temperature gradients lead to incorrect heat flux calculations. These findings reinforce the importance of selecting an appropriate balance between physics-based constraints and available data when applying PINNs to thermal modeling problems.

While PINNs show strong potential in inverse thermal analysis, certain limitations remain. The models are sensitive to high-gradient regions, which can impact prediction accuracy near steep temperature variations. Additionally, PINNs require retraining when conditions change, limiting their direct application in real-time, such as digital twins for machine tools. Future research should explore strategies to improve adaptability, such as integrating physics-informed neural operators or leveraging transfer learning to reduce the computational cost of retraining under varying conditions. Extending this methodology to three-dimensional and transient thermal problems would further clarify the practical advantages and challenges of using PINNs in complex machine tool geometries.

Acknowledgements

The funding for this work was provided initially by the Agencia Estatal de Investigación from the Spanish Government, under project SUSTWINABLE (No. CPP2021-008812), and later by the Basque Country Business Development Agency (SPRI) under the ELKARTEK 2024 MECACOGNIT (KK-2024/00030) project.

References

[1] M. W. M. G. Dissanayake and N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, *Communications in Numerical Methods in Engineering*, vol. **10**, p. 195–201, March 1994. [2] Jain S C, Willander M, Narayan J and van Overstraeten R 2000 *J. Appl. Phys*. **87** 965

[2] 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**, p. 686–707, February 2019.

[3] NAFEMS, "The Standard NAFEMS Benchmarks," NAFEMS Ltd., October 1990

[4] S. Wang, S. Sankaran, H. Wang and P. Perdikaris, An Expert's Guide to Training Physics-informed Neural Networks, arXiv, 2023.