FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Assessment of Physics-Informed Neural Networks for Thermal Distribution in Composite Materials

Fabián Alejandro Alquinta Galarce

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado em Mecânica Computacional

Supervisor: Prof. Dr. José Joaquim da Mota Machado

Second Supervisor: Prof. Dr. Jhonny de Sá Rodrigues

July 21, 2026

Assessment of Physics-Informed Neural Networks for Thermal Distribution in Composite Materials

Fabián Alejandro Alquinta Galarce

Mestrado em Mecânica Computacional

Approved in oral examination by the committee:

President: Prof. Dr. José Joaquim da Mota Machado

Examiner: Prof. Dr. Ricardo Fernando Rodrigues Pinto

Member: Prof. Dr. Jhonny de Sá Rodrigues

July 21, 2026

Resumo

Os materiais compósitos são amplamente utilizados em aplicações de engenharia devido à sua baixa densidade, elevado desempenho mecânico e adaptabilidade a diferentes requisitos estruturais. No entanto, o seu comportamento térmico é mais complexo do que o dos materiais homogéneos, especialmente quando as propriedades dependentes da direção influenciam o processo de transferência de calor. Neste contexto, a previsão precisa e eficiente de campos de temperatura transientes é importante para a análise de estruturas compósitas sujeitas a diferentes condições de carregamento térmico.

Esta dissertação avalia a capacidade de uma Rede Neuronal Informada pela Física paramétrica para prever campos de temperatura transientes numa placa compósita ortotrópica bidimensional sujeita a fluxos de calor impostos e a perdas de calor por convecção. Foi desenvolvido um modelo de elementos finitos em Abaqus para gerar campos de temperatura de referência para diferentes combinações de fluxos de calor aplicados nas fronteiras inferior e esquerda. Estes dados foram combinados com a equação governante de condução de calor transiente, as condições de fronteira e a condição inicial num enquadramento de treino informado pela física.

A Physics-Informed Neural Network (PINN) foi formulada com variáveis adimensionais e recebeu como entradas as coordenadas espaciais, o tempo e os parâmetros dos fluxos de calor impostos. Foi realizado um estudo de sensibilidade fatorial completo para avaliar a influência da profundidade da rede, do número de épocas, da taxa de aprendizagem e da densidade de pontos de colocação na precisão da previsão e no custo computacional. Os modelos treinados foram validados utilizando casos independentes de fluxos de calor aleatórios dentro do intervalo de carregamento prescrito. Além disso, foi realizada uma avaliação de sensibilidade geométrica para analisar a resposta do modelo treinado a alterações nas dimensões da placa.

A configuração com melhor desempenho consistiu em cinco camadas escondidas com 128 neurónios por camada, treinada durante $10,000$ épocas com uma taxa de aprendizagem de 10^{-3} e o nível de colocação C4. Este modelo alcançou um erro percentual absoluto médio de 0,262% no conjunto de validação Random25, com um erro absoluto médio de 0,216 °C e uma raiz do erro quadrático médio de 0,240 °C. Além disso, a comparação temporal realizada em CPU mostrou que a inferência da PINN foi aproximadamente $3{,}4 \times 10^3$ vezes mais rápida do que as simulações em Abaqus.

Os resultados demonstram que uma única PINN paramétrica treinada consegue aproximar com precisão a resposta térmica transiente de uma placa compósita ortotrópica para uma gama de fluxos de calor impostos. A avaliação de sensibilidade geométrica mostrou também que o modelo se manteve preciso para a geometria de referência e para pequenas variações geométricas, embora o erro tenha aumentado quando a geometria se afastou da configuração utilizada durante o treino. Assim, dentro do domínio paramétrico considerado, a abordagem Finite Element Method (FEM)–PINN proposta representa uma boa estratégia de modelação substituta para a previsão térmica rápida e fisicamente consistente em materiais compósitos.

Palavras-chave: Redes Neuronais Informadas pela Física, transferência de calor transiente, materiais compósitos ortotrópicos, método dos elementos finitos, modelação por modelo substituto, Abaqus

Abstract

Composite materials are widely used in engineering applications due to their low density, high mechanical performance, and adaptability to different structural requirements. However, their thermal behaviour is more complex than that of homogeneous materials, especially when direction-dependent properties influence the heat-transfer process. In this context, accurate and efficient prediction of transient temperature fields is important for the analysis of composite structures under different thermal loading conditions.

This thesis evaluates the capability of a parametric Physics-Informed Neural Network (PINN) to predict transient temperature fields in a two-dimensional orthotropic composite plate subjected to imposed heat fluxes and convective heat losses. A finite element model was developed in Abaqus to generate reference temperature fields for different combinations of heat fluxes applied on the bottom and left boundaries. These data were combined with the governing transient heat-conduction equation, boundary conditions, and initial condition in a physics-informed training framework.

The PINN was formulated in dimensionless variables and received the spatial coordinates, time, and imposed heat-flux parameters as inputs. A full factorial sensitivity study was carried out to assess the influence of network depth, number of epochs, learning rate, and collocation-point density on prediction accuracy and computational cost. The trained models were validated using independent random heat-flux cases within the prescribed loading range. In addition, a geometric sensitivity assessment was performed to examine the response of the trained model to changes in the plate dimensions.

The best-performing configuration consisted of five hidden layers with 128 neurons per layer, trained for $10,000$ epochs with a learning rate of 10^{-3} and the C4 collocation level. This model achieved a mean absolute percentage error of 0.262% over the Random25 validation set, with a mean absolute error of 0.216 °C and a root-mean-square error of 0.240 °C. In addition, the CPU-based timing comparison showed that PINN inference was approximately 3.4×10^3 times faster than Abaqus simulations.

The results demonstrate that a single trained parametric PINN can accurately approximate the transient thermal response of an orthotropic composite plate over a range of imposed heat fluxes. The geometric sensitivity assessment also showed that the model remained accurate for the reference geometry and small geometric variations, although its error increased when the geometry departed from the configuration used during training. Therefore, within the considered parametric domain, the proposed Finite Element Method (FEM)–PINN approach represents a good surrogate modelling strategy for fast and physically consistent thermal prediction in composite materials.

Keywords: Physics-Informed Neural Networks, transient heat transfer, orthotropic composite materials, finite element method, surrogate modelling, Abaqus

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me during these two years of my master's studies.

First, I would like to thank my parents, Raul Alquinta and Margarita Galarce, for their constant support and encouragement throughout my academic journey. I am also deeply grateful to Magdalena Marambio for her patience, understanding, and unconditional support during this important stage.

I would also like to thank my supervisors, Prof. Dr. José Machado and Prof. Dr. Jhonny de Sá Rodrigues, for their kindness, patience, guidance, and support. Their advice and willingness to help me resolve doubts were essential throughout the development of this work.

I am also thankful to my friends Danny and Nacho, whom I met in Portugal, for their friendship, companionship, and encouragement during this period.

To all those mentioned here, and to those whom I may not have named individually, thank you for your support, advice, and encouragement throughout this academic stage.

Finally, I would like to acknowledge Agencia Nacional de Investigación y Desarrollo (ANID), from Chile, for the financial support provided through the Becas Chile program.

Fabián Alejandro Alquinta Galarce

Declaração de Honra

I declare that this dissertation is my own work and has not previously been submitted or assessed at this or any other institution. References to other authors (statements, ideas, thoughts), as well as the use of published works, strictly comply with the rules of attribution and are duly identified in the text and in the bibliographic references, in accordance with the applicable referencing standards. I am aware that the practice of plagiarism or self-plagiarism constitutes an academic offence, as established in the U.Porto Code of Ethics.

I further declare that I have included text(s) published in co-authorship, and that I have specified, with transparency and rigor, the contribution of each co-author, as well as stated whether there exists any other dissertation in which the same text(s) may also have been included. The authorization from the journal in which the work is published is also included, together with the express agreement signed by all co-authors authorizing the inclusion of the article.

Fabián Alejandro Alquinta Galarce

Contents

List of Figures

List of Tables

List of Acronyms

PINN Physics-informed neural networks
BC Boundary Condition
CUDA Compute Unified Device Architecture
FEM Finite Element Method
IC Initial Condition
MAE Mean Absolute Error
MAPE Mean Absolute Percentage Error
PDE Partial Differential Equation
PINN Physics-Informed Neural Network
RMSE Root Mean Square Error
ReLU Rectified Linear Unit
AD Automatic Differentiation
ML Machine Learning
POD Proper Orthogonal Decomposition

Chapter 1

Introduction

Composite materials are widely used in engineering applications due to their combination of properties related to low density, high mechanical performance, and adaptability to different structural requirements allow efficient lightweight designs. Although these advantages are often discussed from a structural point of view, thermal effects are also relevant during manufacturing and service, where composite components may experience heating, cooling, imposed heat fluxes, or convective heat losses. In fibre-reinforced composites, this thermal response can be complex because the effective conductivity depends on the properties and arrangement of the fibres and matrix, leading to direction-dependent heat-transfer behaviour [2, 7]. Therefore, predicting transient temperature distributions in composite materials is an important step for analysing thermal problems with anisotropic or orthotropic properties.

Numerical methods such as the Finite Element Method (FEM) are commonly used to solve transient heat-transfer problems, since they can represent material properties, geometry, and Boundary Conditions (BCs) with high reliability. Nevertheless, when several loading conditions must be analysed, the repeated execution of finite element simulations may become computationally expensive.

In this context, Physics-Informed Neural Networks (PINNs) appear as a promising alternative for thermal prediction. By combining reference data with the governing heat-transfer equations, PINNs can approximate physical fields while preserving consistency with the underlying physics. This work assesses the use of a parametric PINN to predict transient thermal distributions in an orthotropic composite plate subjected to imposed heat fluxes and convective heat losses.

This work aims to assess whether a single trained PINN can approximate the thermal response of a composite plate over a prescribed range of imposed heat fluxes, while maintaining good agreement with finite element reference solutions. In addition, this work evaluates whether the trained model can provide faster predictions compared with conventional finite element simulations once the training stage has been completed.

The main objective of this work is to evaluate the capability of a parametric PINN to predict transient temperature fields in an orthotropic composite plate subjected to different heat-flux BCs.

To achieve this goal, the following specific objectives are defined:

1. Define a two-dimensional transient heat-conduction problem for an orthotropic composite plate with imposed heat fluxes and convective heat losses.

2. Develop a finite element reference model in Abaqus to generate temperature fields for different thermal loading conditions.

3. Create a finite element model, based on the already defined problem, to serve as reference, generating the temperature fields for the different thermal loading conditions.

4. Formulate a dimensionless parametric PINN that incorporates the governing heat equation, BCs, and Initial Condition (IC) into the loss function.

5. Train the parametric PINN and collect the training data.

6. Define training metrics, allowing to evaluate the model results with its corresponding finite element results.

7. Based on the computed training metrics, perform a sensitivity study to evaluate the influence of selected training and parameters on prediction accuracy and computational cost.

8. Validate the trained PINN against independent random heat-flux cases and compare its inference time with Abaqus simulations.

Therefore, the central question of this work is whether a parametric PINN can act as an accurate and computationally efficient surrogate model for transient thermal prediction in an orthotropic composite plate.

Firstly, the theoretical basis of neural-network approaches for solving differential equations was reviewed, with particular attention to the work of Lagaris, Likas, and Fotiadis [21]. This work presented an early formulation in which artificial neural networks are used to approximate the solutions of ordinary and partial differential equations. Therefore, it provided a relevant conceptual background for understanding the development of physics-informed neural-network methods.

After this initial stage, the study was extended to PINNs, mainly through the framework proposed by Raissi, Perdikaris, and Karniadakis [26]. The review was then focused on how PINNs have been formulated for heat-transfer problems, particularly in cases where the governing heat equation, the initial condition, and the thermal BCs are included as separate terms of the loss function [8, 38]. This was relevant for the present work because the thermal problem considered here is not defined by a single fixed loading case, but by a range of imposed heat-flux conditions. Previous studies have shown that this type of variability can be addressed by using PINNs as surrogate models and, in some cases, by including problem parameters or process conditions as additional inputs of the neural network [3, 16, 35]. Therefore, this review supported the formulation adopted in this work, where the imposed heat fluxes are treated as input variables together with space and time, allowing a single trained PINN to approximate the transient temperature field for different boundary-condition combinations.

A bibliographic search was carried out to identify suitable thermoplastic composite materials and obtain representative thermal properties for the numerical implementation. The carbon/polyamide 410 thermoplastic composite studied by Asfew, Ivens, and Moens [4] was finally selected, since its thermal properties showed a clear directional dependence. This anisotropic behaviour was valuable for the objectives of the work, as it allowed the proposed methodology to be assessed in a material with different thermal conductivities along the principal directions.

Subsequently, a custom implementation was developed for a two-dimensional heat-transfer problem in an orthotropic composite plate. In the first stage, the PINN was implemented using dimensional variables, namely the spatial coordinates, time, temperature, and imposed heat fluxes expressed in their physical units. Its predictions were compared with finite element results generated in Abaqus through a dedicated script. Once the behaviour of the model was verified, the formulation was adapted to the dimensionless parametric case considered in this work, where the imposed heat fluxes are included as input variables of the neural network.

Finally, the implemented PINN was used in a sensitivity study to evaluate the influence of selected training and architectural parameters, such as the number of epochs, learning rate, and collocation-point density, on prediction accuracy and computational cost. The training process was executed on GPU to record training times, while a separate CPU-based analysis was performed to measure PINN inference time and compare it with the computational time required by Abaqus simulations. This comparison is central to the present work, since it allows the proposed PINN to be assessed not only in terms of its agreement with finite element reference solutions, but also in terms of its potential as a computationally efficient surrogate model for repeated transient thermal simulations.

This work is structured to present the theoretical background, methodology, implementation, and assessment of a parametric PINN for thermal prediction in composite materials. It begins by presenting a State of the Art, where the thermal behaviour of composite materials, conventional numerical methods, Machine Learning (ML) approaches, and PINNs applied to heat-transfer problems are reviewed. This chapter establishes the theoretical basis of the work and identifies the research gap addressed in the study.

The subsequent chapter describes the methodology used to develop the proposed approach. It introduces the two-dimensional orthotropic heat-transfer problem, the finite element reference model in Abaqus, the non-dimensional formulation, the PINN architecture, the loss function, the training procedure, and the validation strategy. The Results chapter then presents the sensitivity study, discusses the influence of the selected hyperparameters, identifies the best-performing configuration, and compares the computational cost of PINN inference with Abaqus simulations. Finally, the Conclusions and Future Work chapter summarises the main findings, discusses the limitations of the proposed approach, and outlines possible extensions of the study.

Chapter 2

State of the Art

This chapter reviews the theoretical and computational background relevant to the use of PINNs for thermal analysis in composite materials. The review first addresses heat transfer in composite materials and conventional numerical modelling approaches. Then, ML and PINNs are discussed, with particular emphasis on their application to heat-transfer problems. Finally, the research gap addressed by the present work is identified.

2.1 Thermal behaviour of composite materials

Composite materials are widely used in engineering applications because they combine low density with high mechanical performance. However, their thermal response is generally more complex than that of homogeneous isotropic materials. In fibre-reinforced polymer composites, the effective thermal behaviour depends on the individual thermal properties of the fibres and matrix, the fibre volume fraction and orientation, the fibre-matrix interface, and microstructural features such as dispersion, defects, porosity, and thermally conductive pathways within the reinforcement architecture [2, 7].

A key characteristic of carbon-fibre-reinforced composites is their directional thermal behaviour. Since carbon fibres and polymer matrices have different thermal conductivities, the resulting composite may exhibit anisotropic or orthotropic heat-transfer characteristics. In particular, the thermal conductivity along the fibre direction is usually different from the conductivity in the transverse or through-thickness directions. This directional behaviour was reported in early experimental studies on carbon-fibre-reinforced composites, where different thermal conductivities were measured parallel and perpendicular to the fibre direction, and has continued to be analysed in later numerical and experimental investigations [25, 27].

A key characteristic of carbon-fibre-reinforced composites is their directional thermal behaviour. Since carbon fibres and polymer matrices have different thermal conductivities, the resulting composite may exhibit anisotropic or orthotropic heat-transfer characteristics. In the particular case of uni-directional composites, the thermal conductivity along the fibre direction is usually different from the conductivity in the transverse or through-thickness directions. This directional behaviour was reported in early experimental studies on carbon-fibre-reinforced composites, where different thermal conductivities were measured parallel and perpendicular to the fibre direction, and has continued to be analysed in later numerical and experimental investigations [25, 27].

The directional dependence of thermal conductivity is especially relevant for thermal simulations. The work of Bard et al. [5] investigated carbon-fibre-reinforced epoxy laminates and showed that fibre volume content affects the thermal conductivity in both the fibre and transverse directions. Their results indicated different trends for heat flow along and transverse to the fibre direction, confirming that a single isotropic conductivity value may be insufficient to represent the thermal behaviour of fibre-reinforced composites.

The thermal response of composite materials is also influenced by temperature. Changes in heat capacity, density, thermal expansion, and thermal conductivity can modify the transient temperature field, particularly when the material is subjected to non-isothermal conditions. In the case of carbon/polyamide 410 composites, Asfew, Ivens, and Moens [4] measured temperature-dependent thermophysical properties and reported that heat capacity, density, coefficient of thermal expansion, and thermal conductivity vary with temperature. These results highlight the importance of selecting representative thermal properties when modelling thermoplastic composite materials.

Previous studies have also shown that the internal structure of the composite affects heat transfer across different length scales. Multiscale modelling of carbon-fibre-reinforced polymer composites has shown that the effective thermal conductivity is influenced by factors such as fibre volume fraction, temperature, fibre–matrix interphase characteristics, matrix porosity, and reinforcement architecture [24]. Therefore, the prediction of temperature fields in composites requires a modelling approach capable of accounting for directional thermal properties and boundary-driven heat flow.

In this work, the composite plate is modelled as an orthotropic material with different thermal conductivities in the x- and y-directions. This assumption provides a simplified but physically meaningful representation of the directional heat-transfer behaviour of fibre-reinforced composites. It also supports the formulation of the transient heat conduction problem used later to assess the ability of PINNs to approximate thermal distributions in composite materials.

2.2 Numerical modelling of heat transfer in composite materials

Numerical modelling is widely used to analyse heat transfer in composite materials, especially when the thermal response is affected by anisotropy, heterogeneous microstructures, complex BCs, or transient loading. For composite materials, analytical solutions are usually limited to simplified geometries and idealised BCs. Therefore, numerical methods provide a practical approach for predicting temperature fields in cases where material directionality and boundary-driven heat flow must be considered.

The FEM is one of the most common approaches for solving heat-transfer problems in composite materials. It allows the governing heat conduction equation to be solved over discretised geometries while incorporating material properties, thermal loads, ICs, and BCs. In the context of composite materials, FEM is particularly useful because orthotropic or anisotropic conductivity can be assigned to represent the directional nature of heat conduction.

Previous studies have used numerical methods to analyse both effective thermal properties and transient temperature distributions in composite materials. For example, Kamiński [17] studied the homogenisation of transient heat-transfer problems in composite materials, showing that effective thermal properties can significantly simplify the numerical analysis of heterogeneous composite structures. Similarly, Nasri et al. [24] applied multiscale FEM modelling to plain-woven carbon-fibre-reinforced polymer composites, comparing numerical predictions of thermal conductivity with analytical models and experimental data reported in the literature.

Finite element models have also been applied to transient heat conduction in composite laminates. Abdelal, Abdalla, and Gürdal [1] investigated the transient thermal response of variable-stiffness composite laminates subjected to partial surface heat flux using the FEM. Their study showed that fibre steering and laminate configuration can influence the temperature field, the maximum temperature, and the time required to reach steady-state conditions. This type of analysis highlights the importance of numerical modelling when the thermal response depends on directional material properties.

In thermoplastic composite processing, numerical heat-transfer models are also used to predict temperature evolution during manufacturing. Sonmez and Hahn [29] developed a finite element model coupling heat transfer and crystallisation kinetics for thermoplastic composite tape placement, while later studies analysed temperature distributions in automated fibre or tape placement processes using different numerical approaches, including finite-difference and finite-element models, with process-representative thermal boundary conditions [9, 11, 30]. These works show that numerical thermal modelling is relevant not only for service conditions, but also for understanding temperature distributions during composite manufacturing.

Despite its robustness, the FEM can become computationally expensive when many simulations are required. This limitation is especially relevant in parametric studies, where different combinations of material parameters, geometries, or thermal loading conditions must be evaluated. Each new loading case usually requires a separate numerical solution, which increases the total computational cost. For this reason, FEM is often suitable as a reference method, but may become inefficient when rapid predictions are needed over a parametric space for identification or control purposes.

In the present work, Abaqus is used as the finite element element simulations solver, setting it as the reference solution for the transient heat-transfer problem. The FEM model provides nodal temperature fields for different heat-flux combinations, which are later used to train and validate the PINN. In this context, the finite element model serves as a reliable numerical baseline, while the PINN is assessed as a potential surrogate model capable of approximating the temperature field with reduced inference cost with an acceptable tolerance with respect to this reference.

2.3 Machine learning approaches for physical modelling

ML has become an increasingly relevant tool for modelling physical systems in computational science and engineering. In this context, ML methods are commonly used to learn relationships between input parameters and system responses from numerical, experimental, or observational data, enabling the construction of fast data-driven models for physical systems [20, 32]. Once trained, these models can provide rapid predictions for new input conditions, which is particularly useful when multiple evaluations of a physical system are required.

Following this data-driven strategy, surrogate modelling can be used to replace computationally expensive numerical models with faster approximations trained from reference data. Typically, a high-fidelity model, such as a finite element simulation, is first evaluated for different combinations of input parameters, and the resulting data are then used to train the ML model. This approach has been applied to thermal problems, where ML-based surrogate models approximate temperature-related responses obtained from finite element simulations [28]. Similar strategies have also been used for finite-element-based physical-field prediction in rapid design applications [12]. Therefore, surrogate models are especially useful in parametric studies, optimisation problems, uncertainty analysis, or real-time assessment.

Neural networks are frequently used for surrogate modelling because of their ability to approximate complex nonlinear mappings. In physical modelling, the inputs may correspond to geometric variables, material properties, loading conditions, or time, while the outputs may represent scalar quantities or complete physical fields. For example, Hesthaven and Ubbiali [15] developed a non-intrusive reduced-order modelling approach for parametrised nonlinear Partial Differential Equations (PDEs), in which neural networks were used to approximate the reduced coefficients obtained from a Proper Orthogonal Decomposition (POD)-based representation of high-fidelity solutions.

Deep learning extends this idea by using neural networks with multiple layers, allowing more complex relationships to be learned from large datasets. In physical modelling, deep neural networks can be used to approximate high-dimensional responses, such as spatial or temporal physical fields. For this reason, deep learning has also been applied to thermal field prediction. Chen et al. [10] developed a deep neural network benchmark for temperature field prediction in heat-source layout problems, while Zhao et al. [37] proposed a surrogate model based on data augmentation and deep transfer learning for temperature field prediction. These studies show that data-driven models can approximate thermal responses and reduce the need for repeated numerical simulations.

Despite these advantages, purely data-driven models present important limitations when applied to physical problems. Their accuracy depends on the information contained in the training data and may decrease when the model is evaluated under conditions that are not well represented during training. In addition, models trained only by minimising the discrepancy between predicted and reference data are not explicitly constrained to satisfy the physical laws governing the system. In long-horizon thermal-field prediction, this lack of physical constraints can lead to error accumulation and reduced adaptability, while incorporating physical information can improve the consistency and reliability of the learned model [31].

This limitation is particularly relevant for problems governed by PDEs, such as heat transfer, where the temperature field must satisfy the heat-conduction equation together with the corresponding initial and boundary conditions. These physical relations can therefore be incorporated into the learning process to guide the model towards predictions that are not only close to the reference data but also consistent with the underlying physics. This idea leads to PINNs, where the residuals of the governing equations and the associated conditions are included directly in the training objective.

2.4 Physics-Informed Neural Networks

PINNs are neural networks trained using real or simulated data, physical laws, or a combination of both. In this approach, the neural network approximates the solution of a physical problem while the governing equations are incorporated directly into the training process. Instead of relying only on reference data, the model is penalised when its predictions violate the differential equation, BCs, or ICs involved into the problem [26].

In a conventional supervised neural network, training is driven by input–output examples, and the model parameters are adjusted to reduce the error between predicted and reference values. This approach allows the network to reproduce patterns contained in the training data, but it does not impose any requirement on the predicted field to satisfy the governing equation of the problem. A PINN modifies this training strategy by adding residual terms to the loss function, so that the optimisation process is guided by both the available data and the physical constraints of the problem.

In this formulation, the unknown solution of the governing differential equation is represented by a neural network. If $u(\mathbf{x},t)$ denotes the physical solution field, the PINN approximation can be written as

$$\hat{u}(\mathbf{x},t) = \mathscr{N}(\mathbf{x},t;\omega) \approx u(\mathbf{x},t), \tag{2.1}$$

where $\mathscr{N}(\cdot;\omega)$ is the neural network and ω represents its trainable parameters. The network is trained so that this approximation satisfies the governing differential equation, together with the prescribed BCs and ICs.

A typical PINN loss function can be expressed as

$$\mathscr{L} = W_{\text{data}}\mathscr{L}_{\text{data}} + W_{\text{PDE}}\mathscr{L}_{\text{PDE}} + W_{\text{BC}}\mathscr{L}_{\text{BC}} + W_{\text{IC}}\mathscr{L}_{\text{IC}}, \tag{2.2}$$

where $\mathscr{L}$data represents the mismatch between the network prediction and the available reference data, $\mathscr{L}$PDE denotes the residual associated with the governing PDE, $\mathscr{L}$BC represents the residual associated with the BCs, and $\mathscr{L}$IC represents the residual associated with the ICs. The coefficients W_{data}, W_{PDE}, W_{BC}, and W_{IC} are penalty weights that control the relative contribution of each residual term to the total loss function.

In a PINN, the unknown solution of the governing differential equation is represented by a neural network. More specifically, the network defines a differentiable approximation of the solution field, which must satisfy the governing equation together with the prescribed BCs and ICs. Therefore, the training process seeks not only to fit available data, when such data exist, but also to obtain a solution that is consistent with the physical problem.

A key component of PINNs is the use of Automatic Differentiation (AD). Since the governing equation generally involves spatial and temporal derivatives of the unknown solution, the neural network approximation must be differentiable with respect to its input variables. AD provides a systematic way to compute these derivatives directly from the network output. The resulting derivatives are then used to evaluate the residual of the governing equation at collocation points distributed inside the computational domain. In this way, the physics-informed terms can be incorporated into the loss function without requiring a conventional numerical mesh [22].

PINNs can be applied to both forward and inverse problems. In forward problems, the governing equation, the ICs, BCs, and model parameters are known, and the objective is to predict the physical field of interest. In inverse problems, part of the system information is unknown, such as material properties, source terms, or boundary parameters, and the neural network is trained using available measurements together with the governing equations to infer the missing quantities [22, 26].

The main advantages of PINNs are their ability to combine data and physical constraints, reduce the dependence on labelled data, approximate continuous fields, and evaluate the solution at arbitrary points in the domain. Their mesh-free nature also makes them attractive for problems where generating or adapting a numerical mesh is costly. In addition, because the governing equations are included in the loss function, PINNs can promote physically consistent predictions compared with purely data-driven neural networks.

However, PINNs also present important limitations. Their training process can be sensitive to the choice of network parameters such as its architecture, learning rate, activation function, loss weights, and collocation-point distribution. The different terms in the loss function may have different magnitudes or convergence rates, which can produce imbalances during training. Gradient-flow pathologies have also been identified as a possible cause of training difficulties in PINNs [33]. Moreover, PINNs may fail to learn the correct physical solution in more complex problems if the optimisation process is not properly controlled [19].

These limitations make systematic assessment essential when PINNs are applied to engineering problems. In the context of this work, the predictive performance of the PINN may depend strongly on the network depth, number of training epochs, learning rate, and collocation-point density. For this reason, a sensitivity study is required to evaluate how these training and architectural parameters affect the accuracy and computational cost of the model.

2.4.1 PINNs for heat-transfer problems

PINNs have been increasingly applied to heat-transfer problems because the governing equations, BCs, and ICs can be incorporated directly into the training objective. In these applications, the neural network is commonly used to approximate the temperature field, while the residual of the heat equation is evaluated through AD at collocation points inside the computational domain.

A relevant contribution in this area was presented by Cai et al. [8], who applied PINNs to several prototype heat-transfer problems involving sparse measurements and unknown thermal BCs. Their work showed that PINNs can be used to reconstruct temperature fields while enforcing the governing heat-transfer equations, even when part of the thermal information is not directly available.

Bowman et al. [6] studied PINN-based solutions of the one-dimensional heat equation, considering insulated and convective BCs and a source-term case associated with laser-tissue interaction. This is relevant for thermal problems because the accuracy of the predicted temperature field strongly depends on how ICs and BCs are represented during training.

Heat conduction in anisotropic materials has also been addressed using PINN-based formulations. Xing, Cheng, and Cheng [36] applied a PINN approach to three-dimensional anisotropic steady-state heat conduction problems. Their study is relevant to the present work because orthotropic composite materials require direction-dependent thermal conductivity, which cannot be fully represented by an isotropic heat-conduction formulation.

In heat-transfer applications, PINNs can be used for both forward and inverse problems. In forward problems, the objective is to predict the temperature distribution from known material properties, ICs, and BCs. In inverse problems, the objective is to estimate unknown quantities, such as thermal properties, heat sources, or boundary parameters, using available temperature data together with the governing heat equation.

Overall, previous studies show that PINNs can approximate thermal fields while enforcing the heat equation and the associated thermal conditions. They have been applied to heat conduction, thermal field reconstruction, anisotropic heat-transfer problems, and problems involving incomplete thermal information. However, the literature also reports challenges related to training stability, accuracy near boundaries, sensitivity to hyperparameters, and the balance between data and physics-informed loss terms.

These aspects are directly related to the present work. The current work applies a parametric PINN to predict the transient thermal distribution in an orthotropic composite plate subjected to imposed heat fluxes and convective heat losses. Unlike studies focused on a single loading case, the present formulation includes the imposed heat fluxes as input parameters, allowing one trained network to approximate the temperature field over a prescribed range of thermal loading conditions.

2.5 PINNs as surrogate models for numerical simulations

PINNs can be used not only as numerical solvers for differential equations, but also as surrogate models for high-fidelity numerical simulations. In this context, a surrogate model aims to approximate the response of a physical system with lower computational cost than the original solver. For heat-transfer problems, the trained model can be used to predict temperature fields for new thermal loading conditions without solving the full numerical problem again.

This idea is particularly relevant when conventional numerical methods such as the FEM are used in parametric studies. In a typical finite element workflow, each new combination of material properties, BCs, heat sources, or loading parameters requires a separate simulation. Although FEM is accurate and robust, the computational cost can increase significantly when several cases must be evaluated. A trained PINN, by contrast, can evaluate new input conditions through a forward pass of the neural network, which is generally much faster than running a complete numerical simulation.

Several studies have explored this idea in heat-transfer applications. Manavi, Becker, and Fattahi [23] proposed an enhanced surrogate-modelling framework for heat conduction problems using a physics-informed neural network approach. Their work shows how the heat equation and thermal BCs can be embedded into the learning process to improve the physical consistency of the surrogate model. Similarly, Go, Lim, and Lee [14] developed a PINN-based surrogate model for a virtual thermal sensor with real-time simulation capability, using limited temperature information to estimate temperature fields or heat fluxes in regions without sensors.

Parametric thermal problems are especially suitable for surrogate modelling because the same governing equation must be solved repeatedly for different input parameters. Koric and Abueidda [18] investigated data-driven and physics-informed deep learning operators for the heat conduction equation with a parametric heat source. Although this approach is based on operator learning rather than a standard PINN architecture, it illustrates the same computational motivation: once trained, a physics-informed model can rapidly predict the solution for new parameter values without requiring an independent numerical simulation for each case.

Despite their potential, PINN-based surrogate models require careful validation. Since the trained network approximates the numerical or physical response over a prescribed input domain, it is recommended that its predictions be compared against trusted reference solutions. In engineering problems, these references are commonly obtained from high-fidelity numerical simulations or experimental measurements. This validation is especially important when the model is evaluated under loading conditions that were not explicitly used during training.

A key trade-off of this approach is the difference between training cost and inference cost. Training a PINN may require significant computational effort because the optimisation process while training involves data terms, physics-informed residuals, BCs, and collocation points. However, once the model has been trained, prediction for new cases can be obtained rapidly. Therefore, PINN-based surrogate models are most useful when the trained model is reused many times, such as in parametric studies, optimisation, design exploration, sensitivity analysis, or real-time prediction.

Overall, the reviewed studies show that PINN-based surrogate models can reduce the computational cost of repeated numerical simulations in parametric heat-transfer problems. Once trained, they can provide rapid predictions for new input conditions, making them useful for design exploration, optimisation, sensitivity analysis, and real-time estimation. However, their performance remains problem-dependent and must be validated against trusted reference solutions, especially under conditions not included during training. Therefore, a reliable PINN-based surrogate model should be assessed in terms of predictive accuracy, computational efficiency, and sensitivity to training and architectural choices.

2.6 Research gaps

The literature reviewed in this chapter shows that the thermal analysis of composite materials requires the consideration of directional material properties, transient heat-transfer behaviour, and boundary-condition effects. Fibre-reinforced polymer composites may exhibit anisotropic or orthotropic thermal conductivity, which makes the prediction of temperature fields more complex than in homogeneous isotropic materials [5, 25, 27]. For this reason, numerical methods such as the FEM remain widely used as reliable tools for simulating heat-transfer problems in composite materials [1, 11, 17].

At the same time, ML and PINNs have been increasingly explored as surrogate modelling tools for physical systems. Neural-network-based approaches have shown potential for fast approximation of parametrised physical responses and thermal fields [10, 15, 37]. PINNs further extend this idea by incorporating the residuals of the governing equations and associated conditions directly into the loss function [22, 26]. Previous studies have shown that PINNs can be applied to heat-transfer problems, thermal field reconstruction, anisotropic heat conduction, and surrogate modelling [6, 8, 23, 36].

Despite these advances, fewer studies have systematically evaluated PINNs as parametric surrogate models for transient thermal distributions in orthotropic composite materials. Many works focus on a specific configuration or a single loading condition, while the effect of training parameters such as network architecture, learning rate, number of epochs, and collocation-point density is not always analysed in detail. This is important because PINN performance can be sensitive to optimisation settings and loss-term balance [19, 33].

The present work addresses this gap by assessing a parametric PINN for the prediction of transient temperature fields in a two-dimensional orthotropic composite plate. The model considers imposed heat fluxes and convective heat losses, and the imposed thermal loads are included as input parameters of the neural network. In this way, a single trained PINN is used to approximate the temperature distribution over a prescribed range of boundary loading conditions.

The contribution of this work is threefold. First, a hybrid FEM–PINN methodology is developed, where finite element simulations in Abaqus provide reference temperature fields and the heat equation is enforced through physics-informed residuals. Second, a full factorial sensitivity study is performed to evaluate the influence of network depth, training epochs, learning rate, and collocation-point density on prediction accuracy and computational cost. Third, the trained PINN is validated against independent random heat-flux cases and its inference time is compared with the computational time required by a validated FEM solver simulations.

The following chapter defines the methodology used to implement this assessment, including the physical problem, finite element reference model, non-dimensional formulation, PINN architecture, loss function, training strategy, validation procedure, and computational-time comparison.

Chapter 3

Materials and Methods

This chapter presents the methodology proposed to assess PINNs for transient thermal prediction in composite materials. The approach will combine reference solutions to be generated with the FEM in Abaqus and a parametric PINN designed to approximate the temperature field under different thermal loading conditions.

The methodology will follow a hybrid numerical–physics-informed framework. First, a two-dimensional transient heat-transfer problem will be defined for an orthotropic composite plate. Then, finite element simulations will be carried out for several heat-flux combinations and used to obtain reference temperature fields. These data will be combined with the governing heat equation, BCs, and IC to train and evaluate a parametric PINN. Finally, a sensitivity study and a validation procedure will be used to assess prediction accuracy and computational cost.

3.1 Methodological framework

The proposed methodology combines a finite element reference model with a physics-informed neural network. Abaqus will be used to solve the transient heat conduction problem for a set of prescribed thermal loading cases, and the resulting temperature fields will provide the reference data for training and validation.

The PINN will be trained to approximate a continuous temperature field while satisfying the governing heat equation and the imposed thermal BCs and IC. Unlike a purely data-driven neural network, its loss function will include physics-informed terms that penalise deviations from the PDE and from the prescribed conditions. Therefore, the model will be assessed not only by its agreement with the finite element data, but also by its ability to reproduce the underlying thermal behaviour.

The complete workflow can be summarised as follows:

1. Define the transient heat-transfer problem, material properties, and imposed thermal conditions.

2. Define the parameters to be varied in the study, including the thermal loading values and their combinations, in order to generate the different simulation cases for both the FEM model and the PINN.

3. Generate finite element reference solutions in Abaqus for the selected parameter combinations.

4. Non-dimensionalise the governing variables and parameters.

5. Train a parametric PINN using data and physics-informed residuals.

6. Perform a full factorial sensitivity study.

7. Validate the resulting models using independent heat-flux cases.

8. Compare the computational cost of PINN inference and Abaqus simulations.

3.2 Physics-Informed Neural Networks

PINNs are neural networks designed to approximate the solution of physical problems by combining data-driven learning with the governing equations of the system. In a conventional supervised neural network, the model is trained by minimising the difference between predicted and reference data. In contrast, a PINN adds physics-informed terms to the loss function, so that the network is penalised when its predictions violate the governing PDE, BCs, or ICs.

In this work, the PINN will be used to approximate the transient temperature field in an orthotropic composite plate. The neural network will receive the spatial coordinates, time, and thermal loading parameters as inputs, and will return the predicted temperature at each queried space–time point. Thus, the temperature field will be represented as a continuous function learned by the network, rather than only as values defined at the finite element mesh nodes.

To impose the heat equation within the PINN loss function, derivatives of the predicted temperature field with respect to the input variables will be required. For the transient heat-transfer problem considered in this work, these will include the time derivative of temperature and its spatial derivatives, such as $\partial T/\partial t$, $\partial^2 T/\partial x^2$, and $\partial^2 T/\partial y^2$. These derivatives will be obtained directly from the neural network output using AD, which will allow the residual of the governing equation to be evaluated at collocation points inside the domain.

The same principle will be used to impose the thermal BCs and the IC. Spatial derivatives of the predicted temperature will be used to evaluate the heat-flux boundary residuals, while the initial condition will be imposed by comparing the predicted temperature at the initial time with the prescribed initial temperature field. These residuals will be incorporated into the training objective together with the finite element temperature samples generated in Abaqus.

As a result, the training process will combine two complementary sources of information: supervised reference data from the FEM model and physics-informed residuals from the governing problem. This hybrid strategy is expected to encourage the PINN to produce predictions that are close to the finite element solution while remaining consistent with the transient heat-conduction behaviour imposed by the PDE, BCs, and IC.

In the present study, the PINN will be formulated as a parametric surrogate model. The imposed heat fluxes on the bottom and left boundaries will be included as input parameters, allowing a single network, once trained, to approximate the thermal response over a range of loading conditions. This formulation will be used to evaluate whether PINNs can provide accurate and computationally efficient temperature-field predictions for composite materials under different thermal scenarios.

3.3 Physical problem definition

The case study consists of the transient thermal response of a two-dimensional orthotropic composite plate subjected to imposed heat fluxes and convective heat losses. The computational domain is a rectangular plate defined as

$$\Omega = \{(x,y) : 0 \leq x \leq L_x,\ 0 \leq y \leq L_y\}, \tag{3.1}$$

where L_x and L_y are the plate dimensions in the x- and y-directions, respectively. The computational domain and the applied thermal boundary conditions are shown in Figure 3.1.

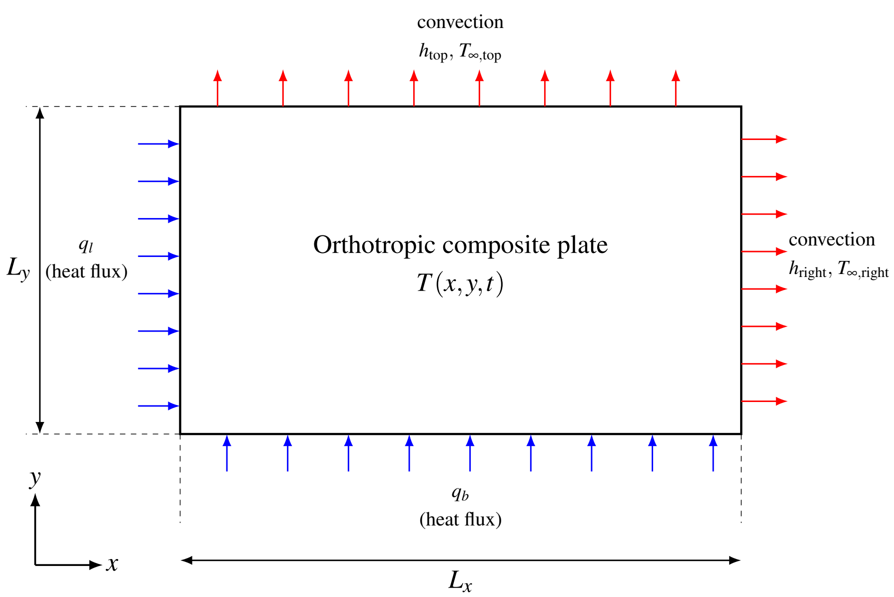


Figure 3.1: Two-dimensional plate geometry and thermal boundary conditions.

The unknown solution of the transient thermal problem is the temperature field $T(x,y,t)$, where x and y are the spatial coordinates and t is time. This function describes the temperature distribution throughout the domain and over the analysed time interval. For the orthotropic material considered in this work, $T(x,y,t)$ must satisfy the transient heat conduction equation

$$\rho c_p \frac{\partial T}{\partial t} = k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2}, \qquad (x,y) \in \Omega,\ t > 0, \tag{3.2}$$

where ρ is the density, c_p is the specific heat capacity, and k_x and k_y are the thermal conductivities along the principal material directions. This formulation accounts for the different conductive behaviour of the composite material in the two spatial directions.

Prescribed heat fluxes are applied along the bottom and left boundaries of the plate. Convective heat losses are imposed along the top and right boundaries. The initial temperature is assumed to be uniform over the entire domain,

$$T(x,y,0) = T_0. \tag{3.3}$$

The material considered in this study is a carbon/polyamide 410 thermoplastic composite. The thermal properties to be used in the model are taken from Asfew, Ivens, and Moens [4]. The longitudinal and transverse thermal conductivities, density, and specific heat capacity correspond to values reported at a temperature of 100°C and will be assumed constant throughout the transient analysis.

The physical and thermal parameters to be used in the case study are summarised in Table 3.1.

Table 3.1: Physical parameters of the transient heat-transfer problem.

Parameter	Symbol	Value
Plate length	L_x	0.075 m
Plate height	L_y	0.040 m
Thermal conductivity in x	k_x	$9.0\ \mathrm{W/(m\,K)}$
Thermal conductivity in y	k_y	$1.0\ \mathrm{W/(m\,K)}$
Density	ρ	$1200\ \mathrm{kg/m^3}$
Specific heat capacity	c_p	$1940\ \mathrm{J/(kg\,K)}$
Initial temperature	T_0	25°C
Top convection coefficient	h_{top}	$25\ \mathrm{W/(m^2\,K)}$
Right convection coefficient	h_{right}	$1\ \mathrm{W/(m^2\,K)}$
Ambient temperature, top	$T_{\infty,\mathrm{top}}$	25°C
Ambient temperature, right	$T_{\infty,\mathrm{right}}$	25°C
Final simulation time	t_{final}	600 s

In this study, only the heat fluxes imposed on the bottom and left boundaries are varied, while the remaining physical and thermal parameters are kept fixed. Therefore, the temperature solution also depends on these boundary loading parameters and can be expressed as

$$T = T(x,y,t;q_b,q_l), \tag{3.4}$$

where q_b and q_l are the imposed heat-flux values on the bottom and left boundaries, respectively. This parametric formulation will allow the PINN to be trained over a range of boundary loading conditions instead of approximating the solution for a single heat-flux case.

3.4 Finite element reference model

Finite element simulations will be carried out in Abaqus to generate the reference temperature fields to be used in this study. The finite element model will reproduce the physical problem described in Section 3.3, including the same geometry, material properties, IC, and thermal BCs.

The model will be implemented as a two-dimensional transient heat-transfer analysis. Heat fluxes q_b and q_l will be prescribed on the bottom and left boundaries, respectively, while convection will be imposed on the top and right boundaries using the heat transfer coefficients and ambient temperatures listed in Table 3.1.

Time integration will use the automatic increment strategy available in Abaqus. The total simulation time will be set to 600 s, with an initial increment of 3 s, a minimum increment of 10^{-8} s, a maximum increment of 150 s, and a maximum number of 10 000 increments. The maximum allowable temperature change per increment will be controlled using $\Delta T_{\max} = 5°\text{C}$.

The plate will be discretised using a structured two-dimensional heat-transfer mesh. The characteristic mesh size will be defined as $L_x/50$, and four-node heat-transfer elements of type DC2D4 will be used. The resulting finite element mesh is shown in Figure 3.2; this discretisation contains 1428 nodes and 1350 elements.

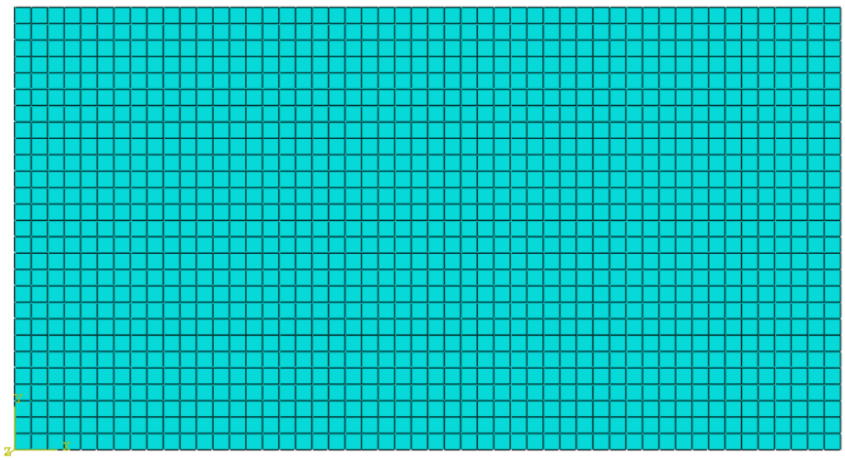


Figure 3.2: Finite element mesh to be used in Abaqus.

The finite element solution will be used for two purposes. First, nodal temperature values will provide supervised data during PINN training. Second, the same finite element framework will provide the reference solutions against which the trained PINN models will be evaluated.

The training data set will consist of 25 predefined thermal loading cases, obtained from the structured combination of five values of the bottom heat flux and five values of the left heat flux,

$$q_b, q_l \in \{5000, 6000, 7000, 8000, 9000\} \text{ W/m}^2. \tag{3.5}$$

Therefore, the training data will cover a regular full factorial grid of $5 \times 5 = 25$ heat-flux combinations. In this design, each value of the bottom heat flux will be combined with each value of the left heat flux, while the geometry, material properties, initial condition, and convective boundary conditions will remain fixed. The resulting Abaqus cases are summarised in Table 3.2.

Table 3.2: Abaqus training cases defined by the heat-flux combinations.

	q_l [W/m^2]				
q_b [W/m^2]	5000	6000	7000	8000	9000
5000	C_1	C_2	C_3	C_4	C_5
6000	C_6	C_7	C_8	C_9	C_{10}
7000	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
8000	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}
9000	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}

Each entry in Table 3.2 represents one Abaqus simulation, defined by a specific pair of heat-flux values (q_b, q_l). For each case, the transient nodal temperature field will be computed over the complete simulation time and then post-processed as reference data for PINN training and assessment.

3.5 Non-dimensional formulation

Before training the PINN, the governing heat-transfer problem will be rewritten in dimensionless form. This step will express the spatial coordinates, time, heat-flux parameters, and temperature field using comparable numerical scales, reducing scale disparities among the physical variables involved in the problem.

The non-dimensionalisation will be performed by introducing reference scales based on the plate dimensions, material properties, imposed heat fluxes, and temperature range defined from the finite element simulations. This scaling improves the conditioning of the PINN training and is consistent with recommendations for stabilising physics-informed neural-network optimisation [34]. It is also compatible with the use of hyperbolic tangent activations, for which poorly scaled inputs may contribute to activation saturation and less favourable gradient propagation [13].

The dimensionless spatial coordinates are defined as

$$\chi = \frac{x}{L_{\text{ref}}} \tag{3.6}$$

$$\psi = \frac{y}{H_{\text{ref}}} \tag{3.7}$$

where $L_{\text{ref}} = L_x$ and $H_{\text{ref}} = L_y$. The dimensionless time is defined as

$$\tau = \frac{k_{\text{ref}}}{\rho c_p L_{\text{ref}}^2}\, t, \tag{3.8}$$

with $k_{\text{ref}} = k_x$. The physical heat-flux intensities considered in this work satisfy

$$q_b, q_l \in [5000, 9000] \ \text{W/m}^2, \tag{3.9}$$

where q_b and q_l denote the bottom and left boundary heat fluxes, respectively. The reference heat flux will be chosen as the maximum imposed heat-flux intensity,

$$q_{\text{ref}} = 9000 \text{ W}/\text{m}^2. \tag{3.10}$$

This value is used only as a scaling factor to define the dimensionless heat-flux intensities

$$\bar{q}_b = \frac{q_b}{q_{\text{ref}}}, \tag{3.11}$$

$$\bar{q}_l = \frac{q_l}{q_{\text{ref}}}. \tag{3.12}$$

Therefore,

$$\bar{q}_b, \bar{q}_l \in \left[\frac{5000}{9000}, \frac{9000}{9000}\right] = [0.5556, 1.0000]. \tag{3.13}$$

The dimensionless temperature will be defined by scaling the dimensional temperature with respect to the initial temperature and a reference temperature range,

$$\theta = \frac{T - T_0}{T_{\text{ref}} - T_0}, \tag{3.14}$$

where T_0 is the initial temperature and T_{ref} is a fixed reference temperature defined from the maximum temperature in the Abaqus finite element reference cases. The corresponding reference temperature difference is

$$\Delta T_{\text{ref}} = T_{\text{ref}} - T_0. \tag{3.15}$$

Once the PINN predicts the dimensionless temperature $\hat{\theta}$, the dimensional temperature will be recovered as

$$\hat{T} = T_0 + \Delta T_{\text{ref}} \hat{\theta}. \tag{3.16}$$

After non-dimensionalisation, the governing equation becomes

$$\frac{\partial \theta}{\partial \tau} = \frac{\partial^2 \theta}{\partial \chi^2} + \beta_y \frac{\partial^2 \theta}{\partial \psi^2}, \tag{3.17}$$

where

$$\beta_y = \frac{k_y}{k_{\text{ref}}} \left(\frac{L_{\text{ref}}}{H_{\text{ref}}}\right)^2. \tag{3.18}$$

The coefficient β_y accounts for both the orthotropic conductivity ratio and the geometric aspect ratio of the plate in the y direction. The reference values and dimensionless parameters to be used in the PINN formulation are summarised in Table 3.3.

Similarly, the physical plate domain is mapped onto the dimensionless computational domain

Table 3.3: Reference values and dimensionless parameters to be used for the PINN formulation.

Parameter	Symbol	Value
Reference length	L_{ref}	0.075 m
Reference height	H_{ref}	0.040 m
Reference conductivity	k_{ref}	9.0 W/(m K)
Heat-flux normalisation scale	q_{ref}	9000 W/m^2
Reference temperature	T_{ref}	242.9°C
Reference temperature difference	ΔT_{ref}	217.9°C
Final dimensionless time	τ_{final}	0.41237
Orthotropic dimensionless parameter	β_y	0.390625

$$\hat{\Omega} = [0,1] \times [0,1], \tag{3.19}$$

which will be used for sampling the collocation points during PINN training.

The same dimensionless temperature definition will also be used to express the initial condition and the thermal boundary conditions in nondimensional form. In particular, the initial condition $T(x,y,0) = T_0$ becomes $\theta(\chi,\psi,0,\bar{q}_b,\bar{q}_l) = 0$. The imposed heat-flux boundary conditions are scaled using the reference heat flux q_{ref}, whereas the convective boundary conditions are expressed in terms of dimensionless Biot numbers and dimensionless ambient temperatures. The detailed derivation of these nondimensional conditions and their corresponding residuals is provided in Appendix A.

3.6 Parametric PINN formulation

A parametric PINN will be implemented to approximate the dimensionless temperature field. The network receives the dimensionless spatial coordinates, dimensionless time, and normalised heat-flux intensities as inputs, and returns the predicted dimensionless temperature,

$$\hat{\theta} = \mathcal{N}(\chi,\psi,\tau,\bar{q}_b,\bar{q}_l;\omega), \tag{3.20}$$

where $\mathcal{N}(\cdot;\omega)$ denotes a fully connected feed-forward neural network with trainable parameters ω, which include all weights and biases.

The input layer contains five variables: χ, ψ, τ, $\bar{q}_b$, and $\bar{q}_l$. The output layer contains one scalar value, corresponding to the predicted dimensionless temperature $\hat{\theta}$. In the sensitivity study, the number of hidden layers will be varied while the hidden-layer width will be kept fixed at 128 neurons. The general architecture is therefore

$$[5,\underbrace{128,\ldots,128}_{n_h},1], \tag{3.21}$$

where n_h is the number of hidden layers. Hyperbolic tangent activation functions will be used in the hidden layers because they provide smooth and differentiable outputs, which are required to compute the derivatives involved in the governing equation residual.

The parametric formulation will allow a single model, after training, to represent the temperature distribution over a range of heat-flux boundary conditions. This differs from a non-parametric formulation, in which a separate model would be required for each loading case.

The physics-informed terms in the loss function will be evaluated at collocation points sampled in the nondimensional computational domain. These points will not be connected as in a finite element mesh; instead, separate point sets will be used to enforce the governing equation, the BCs, and the IC. The corresponding collocation-point sets are illustrated in Figure 3.3.

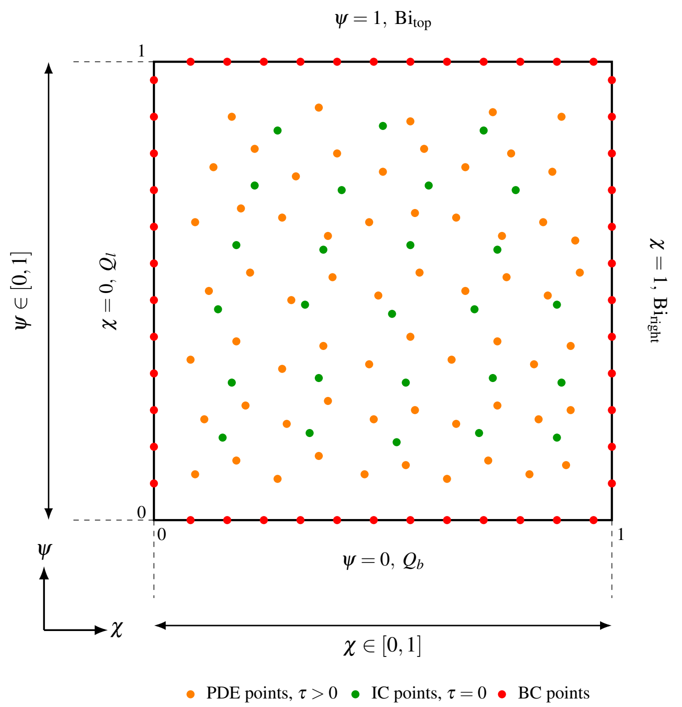


Figure 3.3: Collocation point sets to be used for PINN training in the nondimensional domain.

3.6.1 PINN information flow and automatic differentiation

Before defining the physics-informed residuals, it is useful to describe how information flows through the parametric PINN during training. The objective of the network is to approximate the dimensionless temperature field as a continuous function of the spatial coordinates, time, and heat-flux parameters. For each training point, the input vector is defined as

$$\mathbf{z} = (\chi, \psi, \tau, \bar{q}_b, \bar{q}_l)\,, \tag{3.22}$$

where χ and ψ are the dimensionless spatial coordinates, τ is the dimensionless time, and $\bar{q}_b$ and $\bar{q}_l$ are the normalised heat-flux intensities. The last two quantities act as parameters of the solution, allowing a single neural network to represent the temperature field for different thermal loading conditions.

The parametric PINN maps the input vector $\mathbf{z}$ to the predicted dimensionless temperature field according to

$$\hat{\theta} = \mathcal{N}(\mathbf{z}; \omega), \tag{3.23}$$

where ω denotes the trainable parameters of the neural network, including all weights and biases. In this work, the hidden layers use the hyperbolic tangent activation function. Therefore, for a network with n_h hidden layers, the forward pass can be written as

$$\mathbf{h}^{(0)} = \mathbf{z}, \tag{3.24}$$

$$\mathbf{h}^{(l)} = \tanh\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \qquad l = 1, \ldots, n_h, \tag{3.25}$$

$$\hat{\theta} = \mathbf{W}^{(n_h+1)}\mathbf{h}^{(n_h)} + \mathbf{b}^{(n_h+1)}. \tag{3.26}$$

The use of the $\tanh(\cdot)$ activation function provides a smooth nonlinear approximation of the temperature field, which is important because the physics-informed formulation requires first- and second-order derivatives of the network output with respect to the spatio-temporal variables.

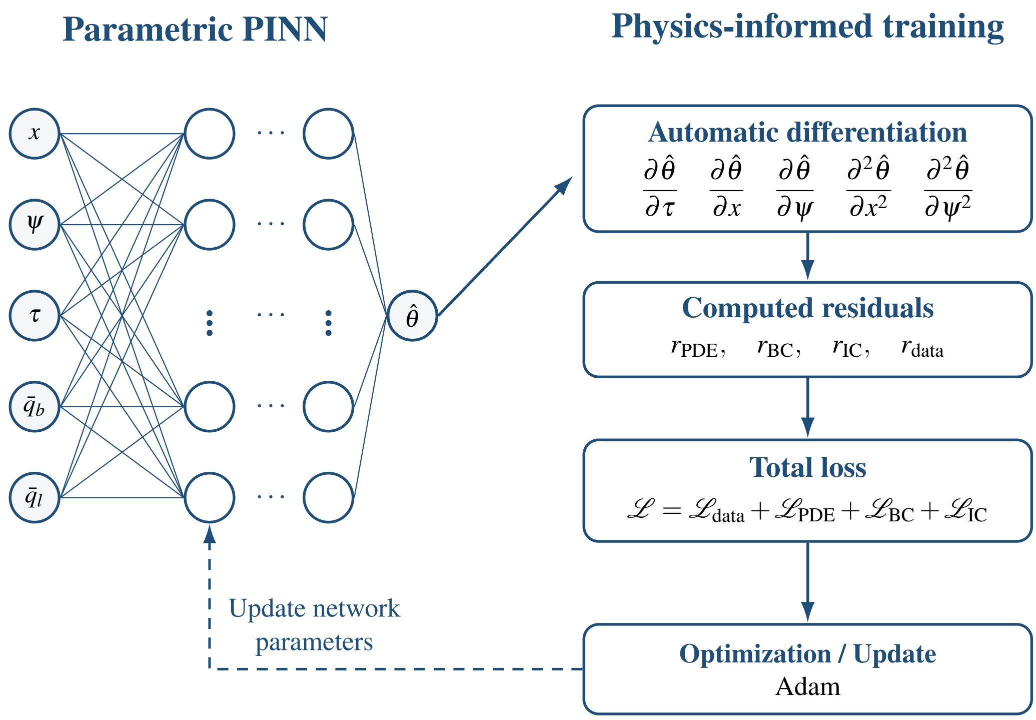


Figure 3.4: Information flow of the parametric PINN from input variables to predicted temperature, automatic differentiation, residual evaluation, loss computation, and parameter update.

During training, the network output will be used in two complementary ways. First, the predicted temperature $\hat{\theta}$ will be directly compared with the dimensionless temperature samples obtained from the finite element reference solutions. This comparison will define the data residual associated with the supervised part of the loss function.

Second, the same predicted field will be differentiated with respect to the spatio-temporal variables using AD. In particular, the physics-informed formulation requires the temporal derivative, the spatial gradients, and the second-order spatial derivatives of the predicted temperature field,

$$\frac{\partial \hat{\theta}}{\partial \tau}, \qquad \frac{\partial \hat{\theta}}{\partial \chi}, \qquad \frac{\partial \hat{\theta}}{\partial \psi}, \qquad \frac{\partial^2 \hat{\theta}}{\partial \chi^2}, \qquad \frac{\partial^2 \hat{\theta}}{\partial \psi^2}. \tag{3.27}$$

These derivatives will be obtained directly from the computational graph of the neural network. Hence, they will not be approximated by finite differences nor computed from a finite element mesh. Instead, AD applies the chain rule through all layers of the neural network, including the tanh activation functions, to provide exact derivatives of the network approximation up to machine precision.

The computed derivatives will then be substituted into the dimensionless heat equation and into the corresponding initial and boundary condition operators. In this way, the PINN will evaluate the residual of the governing PDE at interior collocation points, the residuals of the prescribed heat-flux and convection boundary conditions at boundary collocation points, and the residual of the initial condition at $\tau = 0$. The heat-flux parameters $\bar{q}_b$ and $\bar{q}_l$ will be used to condition the temperature prediction and to define the corresponding boundary residuals for each thermal loading case.

The complete information flow is shown schematically in Figure 3.4. The input variables are first passed through the parametric neural network to obtain $\hat{\theta}$. The predicted temperature and its derivatives will then be used to compute the data, PDE, BC, and IC residuals. These terms are combined into the total loss function,

$$\mathscr{L} = \mathscr{L}_{\text{data}} + \mathscr{L}_{\text{PDE}} + \mathscr{L}_{\text{BC}} + \mathscr{L}_{\text{IC}}, \tag{3.28}$$

which is minimised by updating the trainable parameters ω through the optimiser. This iterative procedure will be repeated until the prescribed number of training epochs is reached.

3.6.2 Physics-informed residuals

The physics-informed residuals are obtained by substituting the nondimensional variables into the dimensional governing equation, initial condition, and boundary conditions. The complete derivation of the PDE, heat-flux, convection, and initial-condition residuals is given in Appendix A.

Substituting the neural-network prediction into the dimensionless heat equation gives

$$r_{\text{PDE}} = \frac{\partial \hat{\theta}}{\partial \tau} - \frac{\partial^2 \hat{\theta}}{\partial \chi^2} - \beta_y \frac{\partial^2 \hat{\theta}}{\partial \psi^2}. \tag{3.29}$$

The residual r_{PDE} will be evaluated at randomly sampled collocation points inside the dimensionless domain. All required spatial and temporal derivatives will be computed using AD. Boundary residuals will also be defined to enforce the imposed heat-flux and convection conditions. On the bottom boundary, $\psi = 0$, the imposed heat flux was enforced through

$$r_{\text{bottom}} = -\frac{k_y}{k_{\text{ref}}}\frac{\partial \hat{\theta}}{\partial \psi} - \bar{q}_b \frac{q_{\text{ref}}\, H_{\text{ref}}}{k_{\text{ref}}\, \Delta T_{\text{ref}}}. \tag{3.30}$$

On the left boundary, $\chi = 0$, the imposed heat flux will be enforced through

$$r_{\text{left}} = -\frac{k_x}{k_{\text{ref}}}\frac{\partial \hat{\theta}}{\partial \chi} - \bar{q}_l \frac{q_{\text{ref}}\, L_{\text{ref}}}{k_{\text{ref}}\, \Delta T_{\text{ref}}}. \tag{3.31}$$

Since $k_{\text{ref}} = k_x$, the factor k_x/k_{ref} is equal to one in the left-boundary residual. It is retained to make the reference-property scaling explicit.

On the top and right boundaries, convective heat transfer will be imposed through dimensionless Biot numbers. The corresponding residuals are

$$r_{\text{top}} = \frac{k_y}{k_{\text{ref}}}\frac{\partial \hat{\theta}}{\partial \psi} + \text{Bi}_{\text{top}}\left(\hat{\theta} - \theta_{\infty,\text{top}}\right), \tag{3.32}$$

and

$$r_{\text{right}} = \frac{k_x}{k_{\text{ref}}}\frac{\partial \hat{\theta}}{\partial \chi} + \text{Bi}_{\text{right}}\left(\hat{\theta} - \theta_{\infty,\text{right}}\right), \tag{3.33}$$

where the Biot numbers are defined as

$$\text{Bi}_{\text{top}} = \frac{h_{\text{top}} H_{\text{ref}}}{k_{\text{ref}}}, \qquad \text{Bi}_{\text{right}} = \frac{h_{\text{right}} L_{\text{ref}}}{k_{\text{ref}}}, \tag{3.34}$$

and the dimensionless ambient temperatures are

$$\theta_{\infty,\text{top}} = \frac{T_{\infty,\text{top}} - T_0}{\Delta T_{\text{ref}}}, \qquad \theta_{\infty,\text{right}} = \frac{T_{\infty,\text{right}} - T_0}{\Delta T_{\text{ref}}}. \tag{3.35}$$

For the present case study, the resulting Biot numbers are

$$\text{Bi}_{\text{top}} = 0.1111, \qquad \text{Bi}_{\text{right}} = 0.00833. \tag{3.36}$$

Since the initial and ambient temperatures are equal to 25°C, the dimensionless ambient temperatures are

$$\theta_{\infty,\text{top}} = 0, \qquad \theta_{\infty,\text{right}} = 0. \tag{3.37}$$

The IC will be imposed as a soft constraint by adding an initial-condition loss term that penalises deviations from the prescribed dimensionless initial temperature. Since $T(x,y,0) = T_0$, the nondimensional initial condition is

$$\theta(\chi, \psi, 0, \bar{q}_b, \bar{q}_l) = 0. \tag{3.38}$$

Therefore, the corresponding residual is

$$r_{\mathrm{IC}} = \hat{\theta}(\chi, \psi, 0, \bar{q}_b, \bar{q}_l). \tag{3.39}$$

3.6.3 Loss function

The total training objective combines supervised finite element data with physics-informed constraints. The loss function is defined as

$$\mathscr{L} = W_{\mathrm{data}}\mathscr{L}_{\mathrm{data}} + W_{\mathrm{PDE}}\mathscr{L}_{\mathrm{PDE}} + W_{\mathrm{BC}}\mathscr{L}_{\mathrm{BC}} + W_{\mathrm{IC}}\mathscr{L}_{\mathrm{IC}}, \tag{3.40}$$

where $\mathscr{L}_{\mathrm{data}}$ is the supervised data loss, $\mathscr{L}_{\mathrm{PDE}}$ is the PDE residual loss, $\mathscr{L}_{\mathrm{BC}}$ is the boundary-condition loss, and $\mathscr{L}_{\mathrm{IC}}$ the IC loss. The coefficients W_{data}, W_{PDE}, W_{BC}, and W_{IC} are weighting factors used to balance the different contributions.

The data loss will be computed as the mean squared error between the PINN prediction and the finite element temperature data. The PDE, BC, and IC losses will be computed as the mean squared values of the corresponding residuals evaluated at collocation points.

The loss weights to be used in the training runs are

$$W_{\mathrm{data}} = 0.05, \qquad W_{\mathrm{PDE}} = 1.0, \qquad W_{\mathrm{BC}} = 1.0, \qquad W_{\mathrm{IC}} = 1.0. \tag{3.41}$$

The data term will be assigned a lower weight than the physics-based terms so that the finite element data guide the solution without dominating the enforcement of the heat equation and boundary conditions.

3.7 Training strategy

The neural network parameters will be optimised using the Adam algorithm with a constant learning rate for each training run. The supervised data will be obtained from the 25 structured heat-flux combinations described in Section 3.4.

Finite element samples will be introduced through shuffled mini-batches. The full finite element data set will be traversed in random order, and a new random permutation will be generated after each complete pass through the data. Interior, boundary, and initial-condition collocation points will be sampled randomly in the dimensionless domain. The heat-flux parameters will be sampled within the normalised range covered by the training cases.

This strategy allows the network to learn from both the finite element reference fields and the governing physical equations. The supervised term provides numerical guidance, while the residual terms constrain the model throughout the spatial, temporal, and parametric domains.

It is important to distinguish between the structured training cases and the cases to be used for performance assessment. The training data will be generated from the 25 predefined heat-flux combinations. The generalisation capability of each trained PINN configuration will then be assessed using a separate set of randomly generated heat-flux cases, as described in Section 3.9.

3.8 Sensitivity study

A full factorial sensitivity study will be conducted to evaluate the influence of network architecture and training hyperparameters on predictive accuracy and computational cost. Four factors will be considered: the number of hidden layers, the number of training epochs, the learning rate, and the collocation-point configuration. The hidden-layer width will be kept fixed at 128 neurons for all configurations.

The factors and their corresponding values are summarised in Table 3.4.

Table 3.4: Factors and levels considered in the sensitivity study.

Factor	Symbol	Values
Hidden layers	n_h	$2,3,4,5,6$
Training epochs	N_{ep}	$6000,7000,8000,9000,10000$
Learning rate	α	$10^{-2},10^{-3},10^{-4},10^{-5},10^{-6}$
Collocation level	C	$C1,C2,C3,C4,C5$

The five collocation-point configurations are detailed in Table 3.5. For each configuration, N_{PDE} denotes the number of interior collocation points, N_{BC} denotes the number of points sampled on each boundary, and N_{IC} denotes the number of initial-condition points. Since the problem has four boundaries, the total number of boundary points is $4N_{\text{BC}}$.

Table 3.5: Collocation-point configurations to be used in the sensitivity study.

Configuration	N_{PDE}	N_{BC} per boundary	Total boundary points	N_{IC}
$C1$	2000	500	2000	1000
$C2$	5000	1250	5000	2500
$C3$	10000	2500	10000	5000
$C4$	15000	3750	15000	7500
$C5$	20000	5000	20000	10000

Combining the five values of the four factors will result in $N_{\text{runs}} = 5 \times 5 \times 5 \times 5 = 625$ independent training configurations. Each configuration will be trained separately, and both its predictive accuracy and training time will be recorded. This will enable the identification of the most accurate configuration, as well as configurations offering favourable accuracy–cost trade-offs.

3.9 Validation and performance assessment procedure

After training, the PINN predictions will be compared against FEM reference solutions. The assessment set will consist of $K = 25$ randomly generated heat-flux cases, referred to as the Random25 set. These cases will be generated within the prescribed heat-flux range using a fixed random seed of 2026, ensuring that all PINN configurations will be evaluated using the same validation cases.

The resulting Random25 validation cases are listed explicitly in Table 3.6. These cases are independent from the structured training cases defined in Table 3.2 and are used only for validation and model ranking.

Table 3.6: Random25 validation heat-flux cases generated with random seed 2026.

Case	q_b [W/m^2]	q_l [W/m^2]	Case	q_b [W/m^2]	q_l [W/m^2]
C01	5715.7	6460.7	C14	8010.9	8601.8
C02	7559.7	5781.6	C15	7060.6	6277.4
C03	6869.1	7379.5	C16	8303.6	7784.0
C04	6482.0	6741.3	C17	6793.5	6255.3
C05	6419.7	6200.0	C18	6355.2	6046.2
C06	8162.1	5837.7	C19	6111.6	7803.4
C07	8620.6	8498.5	C20	5905.3	5911.6
C08	5709.4	8189.8	C21	7103.3	6972.4
C09	7611.1	7426.8	C22	6723.6	7320.1
C10	6193.2	6380.4	C23	7652.7	5755.6
C11	8867.8	8787.3	C24	5051.4	7925.0
C12	8679.4	7253.5	C25	6790.8	7193.9
C13	7543.5	6731.1			

For each validation case, the PINN will be evaluated at the same spatial locations used by the finite element mesh and at the target physical time

$$t = 600 \text{ s}. \tag{3.42}$$

The dimensional PINN prediction at node i of case k will be recovered from the dimensionless network output as

$$\hat{T}_{i,k} = T_0 + \Delta T_{\text{ref}} \hat{\theta}_{i,k}, \tag{3.43}$$

where $\hat{\theta}_{i,k}$ is the corresponding dimensionless prediction. The prediction will then be compared with the finite element reference temperature $T_{i,k}^{\text{FEM}}$.

The nodal temperature error will be defined as

$$e_{i,k} = T_{i,k}^{\text{FEM}} - \hat{T}_{i,k}, \tag{3.44}$$

and the signed percentage error as

$$e_{\%,i,k} = 100\,\frac{T_{i,k}^{\text{FEM}} - \hat{T}_{i,k}}{T_{i,k}^{\text{FEM}}}. \tag{3.45}$$

For each case k, the mean absolute error will be computed as

$$\text{MAE}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} |e_{i,k}|, \tag{3.46}$$

where N_k is the number of finite element nodes in case k. The Root Mean Square Error (RMSE) will be computed as

$$\text{RMSE}_k = \sqrt{\frac{1}{N_k}\sum_{i=1}^{N_k} e_{i,k}^2}. \tag{3.47}$$

The Mean Absolute Percentage Error (MAPE) for each case will be computed as

$$\text{MAPE}_k = \frac{100}{N_k}\sum_{i=1}^{N_k}\left|\frac{T_{i,k}^{\text{FEM}} - \hat{T}_{i,k}}{T_{i,k}^{\text{FEM}}}\right|. \tag{3.48}$$

The case-averaged performance metrics for each trained configuration will be obtained by averaging the case-wise metrics over the $K = 25$ validation cases. The case-averaged MAPE used as the primary ranking criterion will be defined as

$$\overline{\text{MAPE}} = \frac{1}{K}\sum_{k=1}^{K} \text{MAPE}_k, \qquad K = 25. \tag{3.49}$$

Similarly, the case-averaged Mean Absolute Error (MAE) and RMSE will be computed as

$$\overline{\text{MAE}} = \frac{1}{K}\sum_{k=1}^{K} \text{MAE}_k, \tag{3.50}$$

and

$$\overline{\text{RMSE}} = \frac{1}{K}\sum_{k=1}^{K} \text{RMSE}_k. \tag{3.51}$$

The primary ranking criterion for the sensitivity study will be $\overline{\text{MAPE}}$, while $\overline{\text{RMSE}}$ will be used as a secondary criterion when needed. Thus, the best-performing configuration will be identified as the run with the lowest case-averaged MAPE over the Random25 set.

3.9.1 Computational-time assessment

Two computational-time assessments will be carried out. First, the training time of each PINN configuration will be recorded during the sensitivity study. This time will be measured as the wall-clock time required to complete the optimisation process for each configuration and will be stored as the training-loop time. These measurements will be used to compare the training cost of the

625 configurations. All training-time measurements will be obtained using a cloud-based instance equipped with one NVIDIA RTX 5090 GPU.

Second, a separate CPU-based inference-time study will be performed to compare the prediction cost of the trained PINN with the computational cost of Abaqus. This study will be independent of the 625-configuration sensitivity study and will be carried out using the best-performing PINN architecture identified from the validation results.

For the CPU-based comparison, a set of 10 random heat-flux cases will be generated using a fixed random seed of 1999, and each case will be repeated five times for both methods. The PINN inference will be forced to run on CPU by disabling Compute Unified Device Architecture (CUDA) and using a single CPU thread. Abaqus will also be executed on CPU using a single core and with GPU acceleration disabled. The comparison will be performed on a local machine equipped with a 12th Gen Intel(R) Core(TM) i7-12650H processor, with 10 cores and 16 logical processors.

This setup will provide a consistent CPU-based comparison between PINN inference and Abaqus simulations.

Chapter 4

Results

4.1 Sensitivity study results

The sensitivity study comprised a total of 625 PINN configurations, resulting from the full factorial combination of network depth, number of training epochs, learning rate, and collocation-point level defined in Chapter 3. Each configuration was trained independently and then evaluated against the FEM reference solutions corresponding to the Random25 validation set defined in Section 3.9 and listed in Table 3.6.

For each trained configuration, the validation metrics were computed over the same set of 25 independent randomly generated heat-flux cases. The case-averaged MAPE, $\overline{\text{MAPE}}$, over the Random25 set was used as the primary ranking criterion, while the case-averaged RMSE, $\overline{\text{RMSE}}$, was used as a secondary criterion when needed.

For clarity, the trained models are identified in the main discussion by their hyperparameter values: number of hidden layers n_h, hidden-layer width, number of training epochs N_{ep}, learning rate α, and collocation level C. Internal computational identifiers used for post-processing and traceability are reported separately in Appendix B.

4.1.1 Hyperparameter influence

The distribution of case-averaged validation errors, measured using $\overline{\text{MAPE}}$, was further analysed to identify the influence of the main hyperparameters considered in the sensitivity study. Among the tested factors, the learning rate had the strongest effect on predictive accuracy, as shown in Figure 4.1. The error distribution exhibits a clear non-monotonic behaviour with respect to the learning rate. The largest learning rate, $\alpha = 10^{-2}$, produced the highest validation errors, with most configurations concentrated at large $\overline{\text{MAPE}}$ values. This indicates that the optimisation process was likely too aggressive, leading to poor convergence or unstable training. In contrast, the smallest learning rate, $\alpha = 10^{-6}$, also resulted in relatively high errors, suggesting that the parameter updates were too small to achieve sufficient convergence within the prescribed number of epochs.

The best-performing configurations were concentrated around intermediate learning-rate values, particularly $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$. Among these, $\alpha = 10^{-3}$ produced the lowest median error and a compact interquartile range, indicating both high accuracy and relatively consistent performance across the tested architectures, epoch numbers, and collocation levels. Although $\alpha = 10^{-4}$ also yielded low errors, its distribution shows a larger number of upper outliers, indicating that its performance was more sensitive to the remaining hyperparameters. Therefore, the results in Figure 4.1 suggest that $\alpha = 10^{-3}$ provides the most favourable balance between optimisation stability and convergence speed for the present problem.

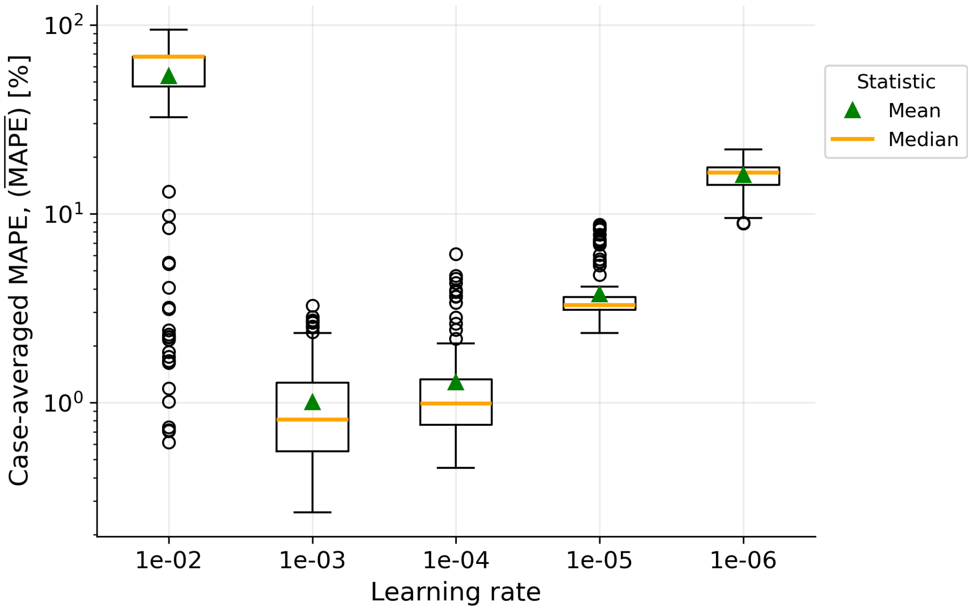


Figure 4.1: Effect of learning rate on $\overline{\text{MAPE}}$.

The effect of network architecture was then analysed for the best-performing learning-rate range. Figure 4.2 shows that increasing the number of hidden layers reduced the case-averaged validation error only up to a certain point. The best results were obtained for intermediate-to-deeper architectures, while the deepest network did not systematically produce lower validation errors. In particular, the best configuration identified in the sensitivity study corresponded to five hidden layers with 128 neurons per layer.

The relationship between $\overline{\text{MAPE}}$ and training time was then analysed to assess the accuracy–cost trade-off of the 625 configurations. As shown in Figure 4.3, configurations trained with $\alpha = 10^{-2}$ and $\alpha = 10^{-6}$ produced the largest errors, indicating unstable optimisation or insufficient convergence. The lowest-error configurations were concentrated around $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$, with the three best results all obtained using $\alpha = 10^{-3}$.

Among the three labelled candidates in Figure 4.3, configuration 3 was discarded because it had both the highest error and the largest training time. The final comparison was therefore between configurations 1 and 2. Although configuration 2 required less training time, configuration

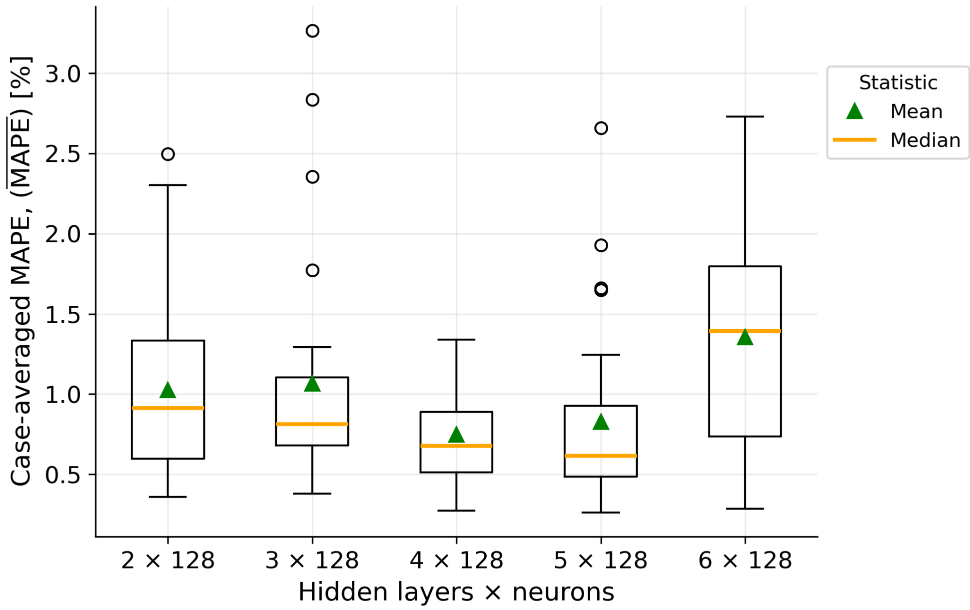


Figure 4.2: Effect of network architecture on $\overline{\text{MAPE}}$ for $\alpha = 10^{-3}$.

1 was retained because it achieved the lowest $\overline{\text{MAPE}}$.

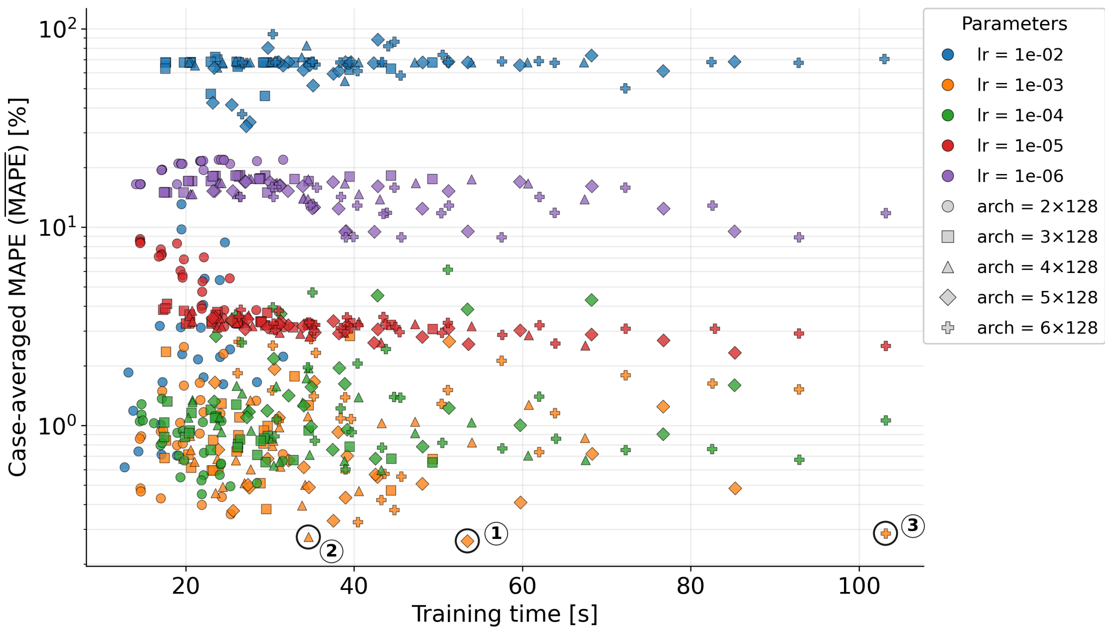


Figure 4.3: Accuracy–cost trade-off coloured by learning rate and marked by architecture.

To further interpret the selected candidates, the same results were represented in terms of collocation level and number of training epochs. Figure 4.4 shows that the three lowest-error configurations were all trained for 10000 epochs. Among them, configuration 3 corresponds to C5 and was again discarded due to its larger error and training time. Configurations 1 and 2 correspond to C4 and C3, respectively. Although the C3 collocation level was faster to train, the C4 collocation level achieved the lowest $\overline{\text{MAPE}}$. Therefore, the candidate labelled as configuration 1 in Figures 4.3 and 4.4 was selected as the best-performing model. This model corresponds to internal run identifier `run_0484`.

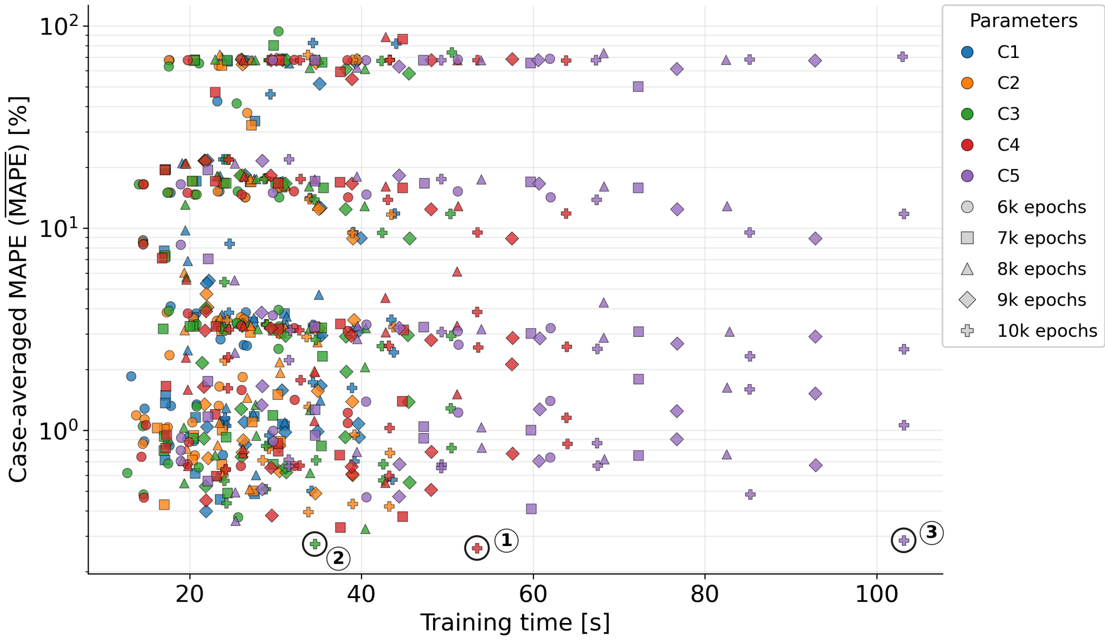


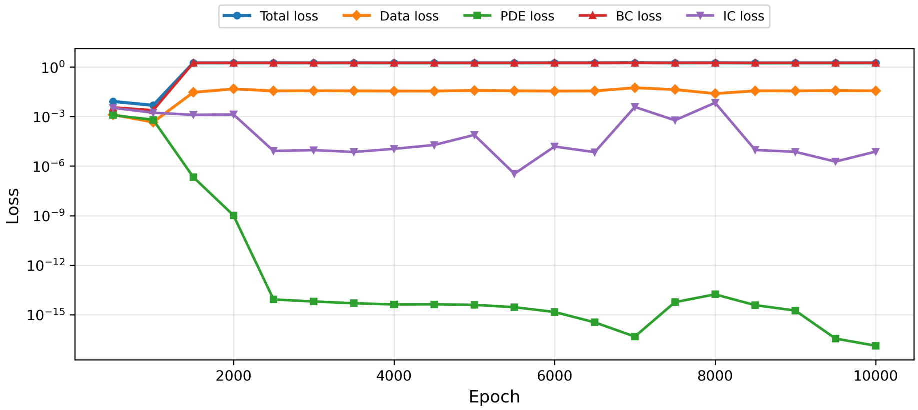
Figure 4.4: Accuracy–cost trade-off coloured by collocation level and marked by training epochs.

4.1.2 Training-loss behaviour

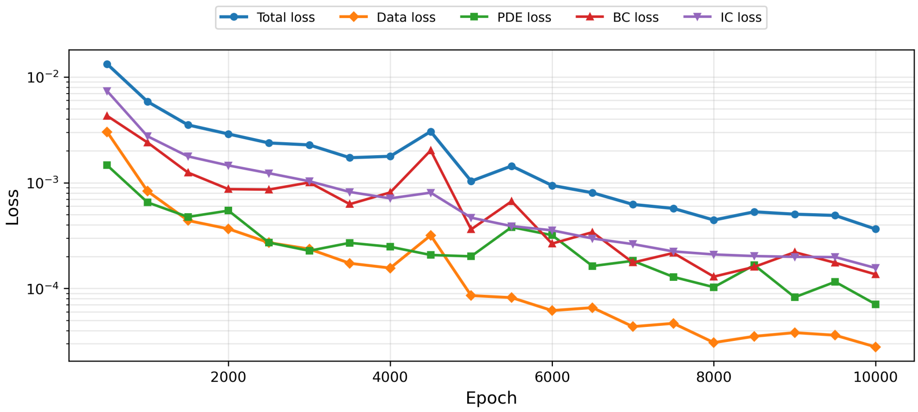
To further analyse the training behaviour associated with the learning-rate effect, the convergence of the training-loss components was examined for three representative configurations. These cases were selected with the same architecture, number of epochs, and collocation level, namely five hidden layers with 128 neurons per hidden layer, $N_{\text{ep}} = 10000$, and collocation level C4. Therefore, the comparison isolates the effect of the learning rate.

The selected cases correspond to three learning-rate values: a high learning rate, $\alpha = 10^{-2}$; the learning rate associated with the best-performing configuration, $\alpha = 10^{-3}$; and a low learning rate, $\alpha = 10^{-6}$. The corresponding internal run identifiers are `run_0479`, `run_0484`, and `run_0499`, respectively.

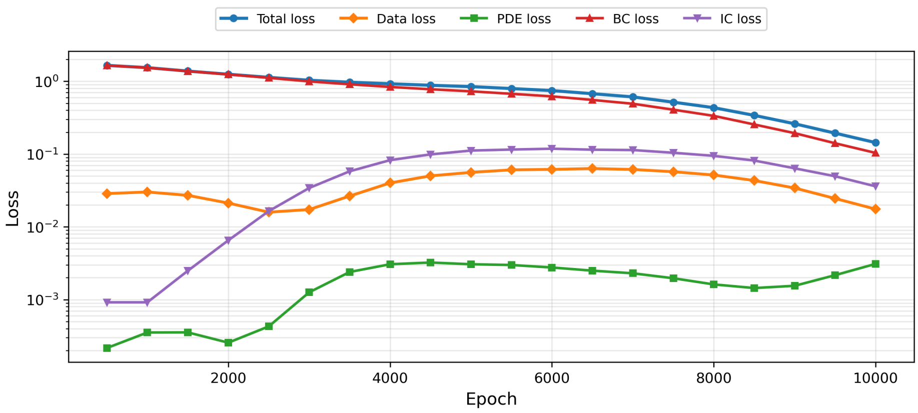
The following figures include the total training loss, the supervised data loss, and the physics-informed residual losses associated with the PDE, BCs, and IC. This allows the optimisation behaviour to be analysed in terms of both data fitting and physics enforcement.



(a)



(b)



(c)

Figure 4.5: Comparison of training-loss convergence for three representative learning rates using the same architecture, number of epochs, and collocation level. (a) $\alpha = 10^{-2}$, internal run identifier `run_0479`. (b) $\alpha = 10^{-3}$, internal run identifier `run_0484`. (c) $\alpha = 10^{-6}$, internal run identifier `run_0499`.

Figure 4.5 shows clear differences in the optimisation behaviour produced by the learning rate. For the high-learning-rate case, $\alpha = 10^{-2}$, the PDE residual is reduced, but the BC loss remains large, indicating that the optimisation does not achieve a balanced minimisation of all loss components.

For the best-performing configuration, `run_0484`, the total loss, data loss, and physics-informed residual losses decrease in a more balanced manner. In particular, the PDE, BC, and IC contributions are simultaneously reduced, suggesting that this learning rate provides a stable compromise between fitting the finite element data and enforcing the governing physical constraints.

For the low-learning-rate case, $\alpha = 10^{-6}$, the loss components evolve more slowly. Although the optimisation is more stable than in the high-learning-rate case, the reduction of the total loss and residual terms is limited within the prescribed number of epochs. This slower convergence is consistent with the higher validation errors obtained for very small learning rates in the sensitivity study.

4.1.3 Best-performing configuration

Table 4.1 summarises the best-performing PINN configuration identified in the sensitivity study. The lowest case-averaged MAPE, $\overline{\text{MAPE}}$, was obtained for a network with five hidden layers and 128 neurons per hidden layer, trained for 10000 epochs with a learning rate of 10^{-3} and the C4 collocation level. The internal computational identifier associated with this trained model is reported separately in Appendix B for traceability.

Table 4.1: Best-performing PINN configuration identified in the sensitivity study.

Parameter	Value
Number of hidden layers	5
Hidden-layer width	128
Number of epochs	10000
Learning rate	10^{-3}
Collocation level	C4
N_{PDE}	15000
N_{BC} per boundary	3750
N_{IC}	7500
Case-averaged MAE, $\overline{\text{MAE}}$ [°C]	0.216
Case-averaged RMSE, $\overline{\text{RMSE}}$ [°C]	0.240
Case-averaged MAPE, $\overline{\text{MAPE}}$ [%]	0.262
Maximum nodal absolute percentage error [%]	1.458
Maximum nodal absolute temperature error [°C]	1.280
GPU training time [s]	53.454

The best-performing model achieved a case-averaged MAPE of $\overline{\text{MAPE}} = 0.262\%$ over the Random25 validation set. In dimensional terms, the corresponding case-averaged MAE and RMSE were $\overline{\text{MAE}} = 0.216$ °C and $\overline{\text{RMSE}} = 0.240$ °C, respectively. These values indicate that the trained PINN was able to reproduce the finite element temperature fields with small average deviations across the random heat-flux cases.

The maximum nodal absolute percentage error obtained for this configuration was 1.458%, while the maximum nodal absolute temperature error was 1.280 °C. This shows that, even in the most demanding pointwise comparison within the Random25 assessment, the discrepancy between the PINN prediction and the FEM reference solution remained limited.

4.2 Validation of the best-performing PINN

After identifying the best-performing configuration described in Table 4.1, a more detailed validation analysis was carried out. This section focuses only on this trained model and examines its performance across the Random25 validation cases. The analysis considers both the final physical time, $t = 600$ s, and a set of intermediate times in order to assess the temporal evolution of the prediction error.

4.2.1 Case-wise error distribution over Random25

Although the case-averaged validation metrics provide a global measure of performance, it is also important to analyse the error distribution across the individual random heat-flux cases. The Random25 validation set was generated using a fixed random seed equal to 2026, producing 25 random combinations of bottom and left heat fluxes within the validation range. Therefore, each case label C01–C25 corresponds to a specific pair of imposed heat fluxes. These heat-flux combinations, together with the final-time case-wise error at $t = 600$ s, are summarised in Table 4.2.

Table 4.2: Random25 validation heat-flux combinations and final-time case-wise error at $t = 600$ s for the best-performing PINN.

Case	q_b [W/m^2]	q_l [W/m^2]	MAPE$_k$ [%]	Case	q_b [W/m^2]	q_l [W/m^2]	MAPE$_k$ [%]
C01	5715.7	6460.7	0.230	C14	8010.9	8601.8	0.335
C02	7559.7	5781.6	0.186	C15	7060.6	6277.4	0.170
C03	6869.1	7379.5	0.338	C16	8303.6	7784.0	0.109
C04	6482.0	6741.3	0.236	C17	6793.5	6255.3	0.142
C05	6419.7	6200.0	0.107	C18	6355.2	6046.2	0.173
C06	8162.1	5837.7	0.248	C19	6111.6	7803.4	0.183
C07	8620.6	8498.5	0.509	C20	5905.3	5911.6	0.454
C08	5709.4	8189.8	0.071	C21	7103.3	6972.4	0.334
C09	7611.1	7426.8	0.256	C22	6723.6	7320.1	0.334
C10	6193.2	6380.4	0.139	C23	7652.7	5755.6	0.196
C11	8867.8	8787.3	0.806	C24	5051.4	7925.0	0.304
C12	8679.4	7253.5	0.180	C25	6790.8	7193.9	0.341
C13	7543.5	6731.1	0.262				

Figure 4.6 shows the corresponding case-wise MAPE, MAPE$_k$, obtained by the best-performing configuration at the final physical time, $t = 600$ s. The cases are kept in the order generated by the fixed random seed, rather than sorted by error, to preserve traceability between the heat-flux combinations, the error metrics, and the subsequent contour visualisations.

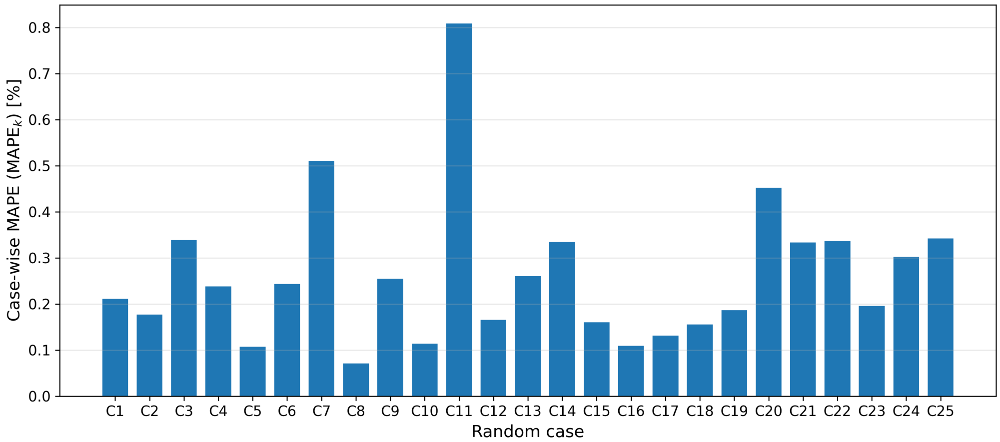


Figure 4.6: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 600$ s.

At the final physical time, the case-wise results show that the error remains consistently low across the Random25 validation set. Most cases present MAPE$_k$ values below approximately 0.35%, while the largest error occurs in case C11, with MAPE$_k \approx 0.81\%$. Since even the worst case remains below 1%, the results indicate that the best-performing PINN provides accurate final-time predictions for all validation cases. The higher errors observed in cases such as C07, C11, and C20 suggest that some random heat-flux combinations are more demanding. Nevertheless, the overall distribution confirms that the low case-averaged error, $\overline{\text{MAPE}} = 0.262\%$, reflects consistently accurate final-time predictions rather than the averaging of isolated favourable cases.

To complement the final-time analysis, the same case-wise error metric was evaluated at intermediate physical times. Figures 4.7–4.12 show the case-wise MAPE distributions obtained at $t = 0$, 100, 200, 300, 400, and 500 s.

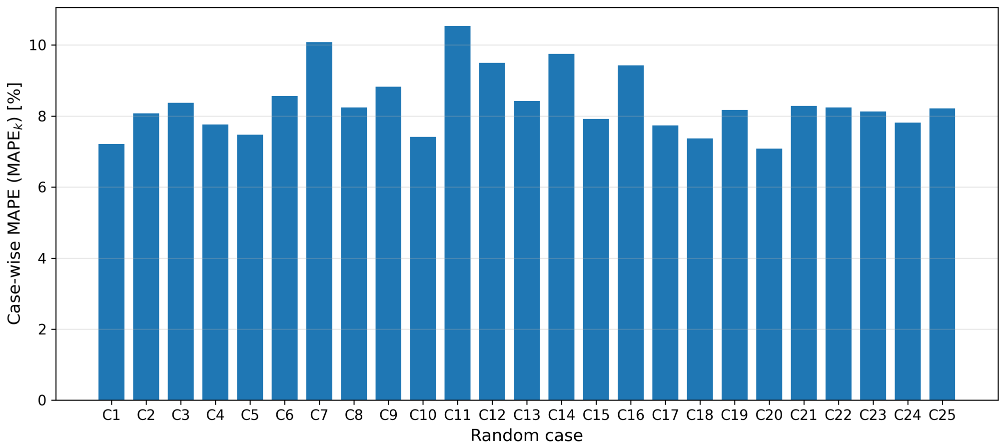


Figure 4.7: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 0$ s.

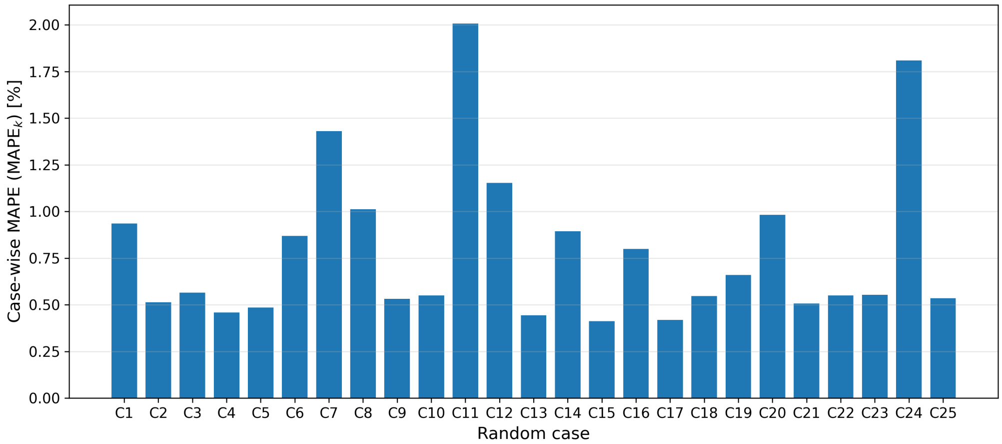


Figure 4.8: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 100$ s.

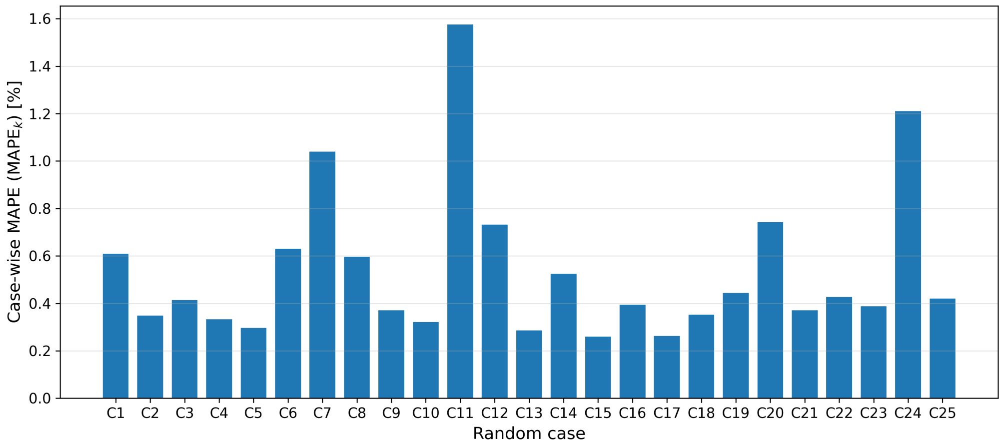


Figure 4.9: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 200$ s.

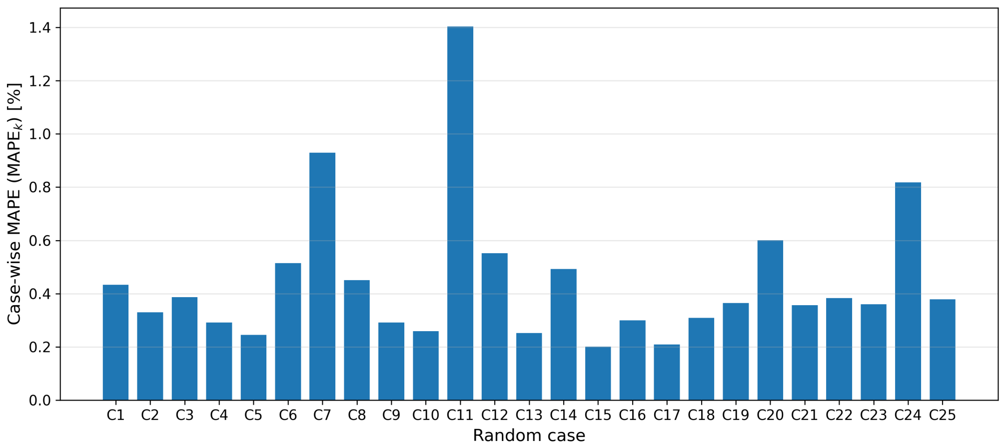


Figure 4.10: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 300$ s.

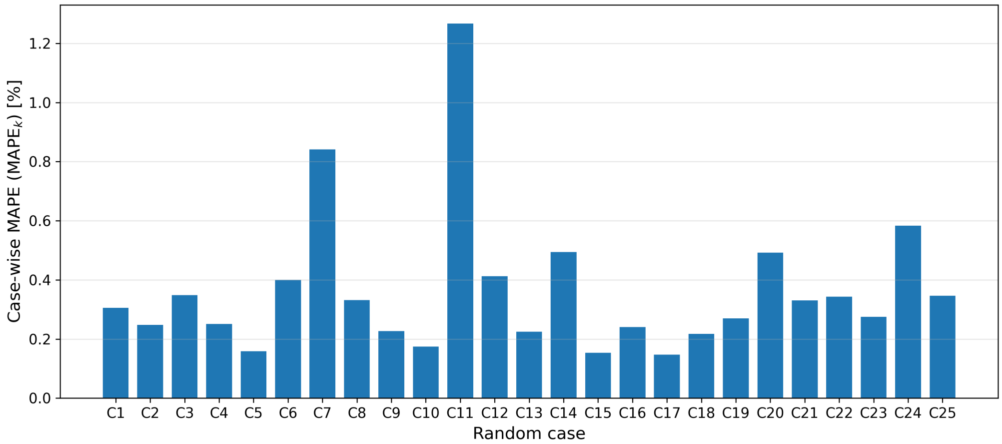


Figure 4.11: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 400$ s.

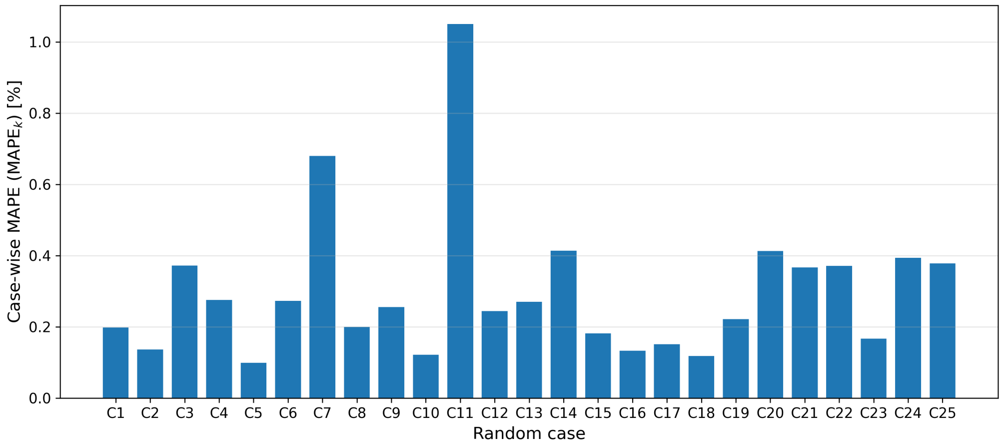


Figure 4.12: Case-wise MAPE, MAPE$_k$, for the best-performing configuration over the Random25 validation cases at $t = 500$ s.

The time-resolved results show a clear reduction of the case-wise error as the transient response evolves. At $t = 0$ s, the relative error is noticeably larger, with MAPE_k values of the order of several percent. This behaviour is consistent with the fact that the initial condition is enforced through the loss function rather than imposed exactly by construction. After the initial instant, the error decreases rapidly: by $t = 100$ s the case-wise errors are mostly close to or below 1%, and from $t = 200$ s onward the distributions remain below approximately 1.6% for all cases. The same cases, particularly C07, C11, and C24, tend to remain among the more demanding cases at intermediate times, indicating that the error pattern is linked to the imposed heat-flux combination.

Overall, the temporal analysis confirms that the best-performing PINN becomes increasingly accurate as the transient thermal field develops. The final-time distribution at $t = 600$ s remains the main quantitative reference for Table 4.2, while the additional time-resolved plots provide evidence that the model maintains a consistent error structure throughout the transient response.

4.2.2 Temporal contour assessment of selected Random25 cases

To complement the global and case-wise validation metrics, the spatial distribution of the PINN prediction error was examined at selected physical times. This analysis provides a more detailed interpretation of how the trained model reproduces the transient thermal field beyond the final-time metrics used for the main ranking procedure.

The Random25 validation set was generated using a fixed random seed equal to 2026, which defined the random combinations of imposed bottom and left heat fluxes reported in Table 4.3. Based on the Random25 error distribution, three cases were selected for spatial contour visualisation. Case C08 was selected as a low-error case, with $q_b = 5709.4\ \mathrm{W/m^2}$ and $q_l = 8189.8\ \mathrm{W/m^2}$. Case C13 was selected as a representative case because its final-time MAPE is close to the case-averaged MAPE of the best-performing model, with $q_b = 7543.5\ \mathrm{W/m^2}$ and $q_l = 6731.1\ \mathrm{W/m^2}$. Case C11 was selected as a high-error and high-load case, with both imposed heat fluxes close to the upper limit of the validation range, namely $q_b = 8867.8\ \mathrm{W/m^2}$ and $q_l = 8787.3\ \mathrm{W/m^2}$.

For each selected case, the PINN prediction, the Abaqus reference solution, and the nodal absolute percentage error are compared at $t = 0$, 200, 400, and 600 s. These contour maps are used to assess the spatial agreement between both solutions and to identify the regions where the prediction error is concentrated.

It should be noted that the final-time metrics at $t = 600$ s are the ones used for the main sensitivity-study ranking. The additional time levels, including $t = 0$, are included here to inspect the transient evolution of the prediction error and to verify the consistency of the learned temperature field throughout the simulation interval.

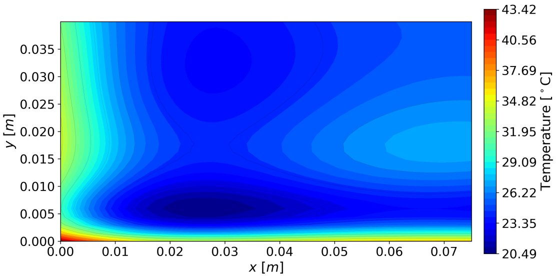
Table 4.3 summarises the error metrics of the Random25 validation cases at the four evaluated physical times. This table extends the final-time validation analysis by showing how the prediction error evolves throughout the transient simulation.

Table 4.3: Exact-time validation errors for the Random25 cases used before contour visualisation.

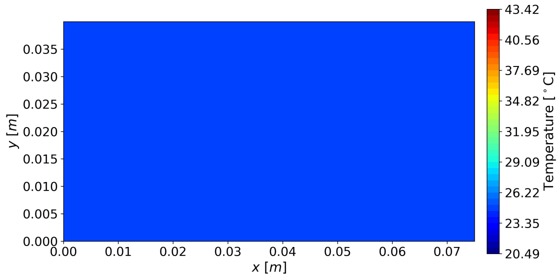
Case	q_b [W/m^2]	q_l [W/m^2]	$t = 0$ s RMSE [°C]	$t = 0$ s MAPE [%]	$t = 0$ s RMSE$_\theta$ [$\times 10^{-3}$]	$t = 200$ s RMSE [°C]	$t = 200$ s MAPE [%]	$t = 200$ s RMSE$_\theta$ [$\times 10^{-3}$]	$t = 400$ s RMSE [°C]	$t = 400$ s MAPE [%]	$t = 400$ s RMSE$_\theta$ [$\times 10^{-3}$]	$t = 600$ s RMSE [°C]	$t = 600$ s MAPE [%]	$t = 600$ s RMSE$_\theta$ [$\times 10^{-3}$]
C01	5715.7	6460.7	2.857	7.213	13.110	0.340	0.609	1.558	0.281	0.308	1.289	0.231	0.230	1.062
C02	7559.7	5781.6	3.325	8.073	15.260	0.200	0.349	0.916	0.168	0.248	0.771	0.178	0.186	0.818
C03	6869.1	7379.5	3.283	8.369	15.070	0.194	0.414	0.889	0.208	0.345	0.955	0.275	0.338	1.262
C04	6482.0	6741.3	3.079	7.766	14.130	0.167	0.333	0.767	0.151	0.251	0.692	0.200	0.236	0.920
C05	6419.7	6200.0	2.989	7.473	13.720	0.179	0.296	0.820	0.124	0.158	0.569	0.092	0.107	0.422
C06	8162.1	5837.7	3.591	8.561	16.480	0.326	0.630	1.498	0.257	0.400	1.177	0.239	0.248	1.098
C07	8620.6	8498.5	4.060	10.084	18.630	0.598	1.040	2.745	0.613	0.830	2.815	0.503	0.509	2.306
C08	5709.4	8189.8	3.195	8.240	14.660	0.383	0.596	1.757	0.293	0.328	1.343	0.079	0.071	0.363
C09	7611.1	7426.8	3.515	8.826	16.130	0.192	0.371	0.880	0.146	0.226	0.670	0.223	0.256	1.025
C10	6193.2	6380.4	2.952	7.416	13.550	0.196	0.321	0.901	0.138	0.172	0.635	0.118	0.139	0.539
C11	8867.8	8787.3	4.230	10.531	19.410	0.839	1.575	3.848	0.907	1.253	4.163	0.795	0.806	3.650
C12	8679.4	7253.5	3.945	9.501	18.100	0.425	0.732	1.951	0.303	0.401	1.393	0.192	0.180	0.882
C13	7543.5	6731.1	3.406	8.426	15.630	0.143	0.286	0.657	0.143	0.222	0.654	0.228	0.262	1.046
C14	8010.9	8601.8	3.836	9.749	17.600	0.407	0.524	1.866	0.421	0.494	1.930	0.358	0.335	1.642
C15	7060.6	6277.4	3.191	7.915	14.650	0.134	0.260	0.613	0.105	0.151	0.483	0.150	0.170	0.688
C16	8303.6	7784.0	3.820	9.429	17.530	0.308	0.394	1.415	0.203	0.241	0.931	0.115	0.109	0.528
C17	6793.5	6255.3	3.104	7.733	14.250	0.139	0.263	0.639	0.103	0.145	0.470	0.126	0.142	0.577
C18	6355.2	6046.2	2.953	7.371	13.550	0.208	0.352	0.953	0.168	0.217	0.772	0.161	0.173	0.739
C19	6111.6	7803.4	3.180	8.175	14.590	0.254	0.443	1.167	0.181	0.268	0.831	0.154	0.183	0.706
C20	5905.3	5911.6	2.824	7.083	12.960	0.371	0.742	1.705	0.367	0.491	1.686	0.389	0.454	1.784
C21	7103.3	6972.4	3.290	8.281	15.100	0.168	0.371	0.771	0.199	0.330	0.915	0.276	0.334	1.265
C22	6723.6	7320.1	3.234	8.244	14.840	0.201	0.426	0.923	0.204	0.340	0.937	0.271	0.334	1.243
C23	7652.7	5755.6	3.361	8.132	15.420	0.218	0.388	0.998	0.186	0.275	0.854	0.182	0.196	0.834
C24	5051.4	7925.0	3.052	7.812	14.000	0.613	1.210	2.811	0.515	0.583	2.364	0.302	0.304	1.384
C25	6790.8	7193.9	3.232	8.212	14.830	0.195	0.420	0.895	0.206	0.344	0.945	0.277	0.341	1.269

Figures 4.13–4.24 present the spatial contour comparisons for the three selected Random25 cases at $t = 0$, 200, 400, and 600 s. Each figure compares the PINN temperature field, the Abaqus reference solution, and the nodal absolute percentage error, while the fourth panel reports the heat-flux values and error metrics for the corresponding case and time level.

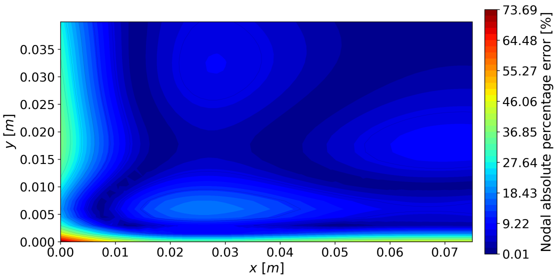
The three cases were selected to represent different validation conditions: C08 as a low-error case, C13 as a representative case, and C11 as a high-load and high-error case.



(a) PINN temperature field.



(b) Abaqus reference field.

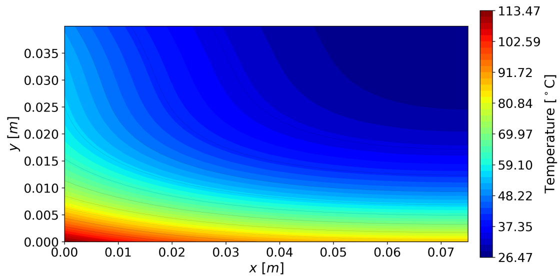


(c) Nodal absolute percentage error.

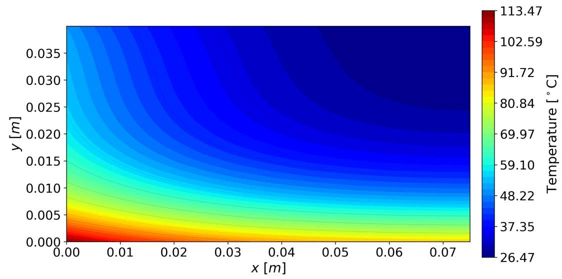
$$q_b = 5709.4\ \mathrm{W/m^2}$$
$$q_l = 8189.8\ \mathrm{W/m^2}$$
$$t = 0\ \mathrm{s}$$
$$T_0 = 25\ ^\circ\mathrm{C}$$
$$\mathrm{RMSE} = 3.195\ ^\circ\mathrm{C}$$
$$\mathrm{MAPE} = 8.240\%$$
$$\mathrm{RMSE}_\theta = 1.466 \times 10^{-2}$$

(d) Case parameters and error metrics.

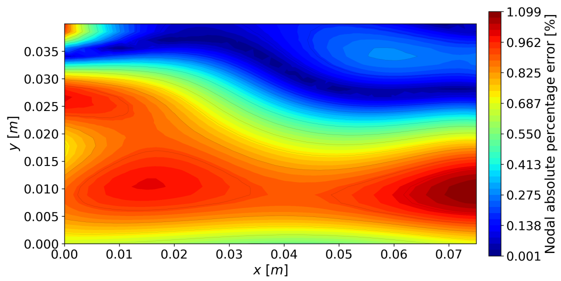
Figure 4.13: Dimensional contour comparison for case C08 at $t = 0$ s.



(a) PINN temperature field.



(b) Abaqus reference field.



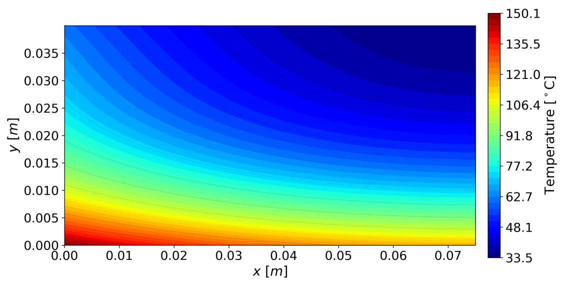
(c) Nodal absolute percentage error.

$q_b = 5709.4\ \mathrm{W/m^2}$
$q_l = 8189.8\ \mathrm{W/m^2}$
$t = 200\ \mathrm{s}$
$T_0 = 25\ °\mathrm{C}$

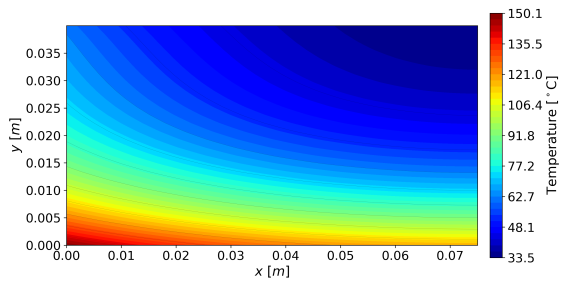
$\mathrm{RMSE} = 0.383\ °\mathrm{C}$
$\mathrm{MAPE} = 0.596\%$
$\mathrm{RMSE}_\theta = 1.757 \times 10^{-3}$

(d) Case parameters and error metrics.

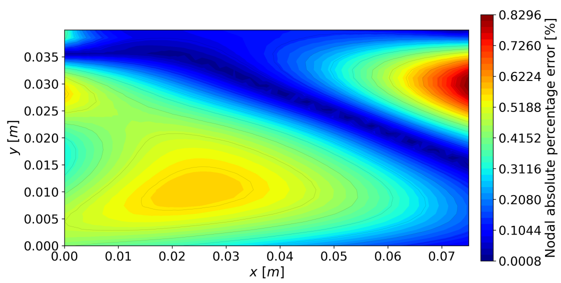
Figure 4.14: Dimensional contour comparison for case C08 at $t = 200$ s.



(a) PINN temperature field.



(b) Abaqus reference field.



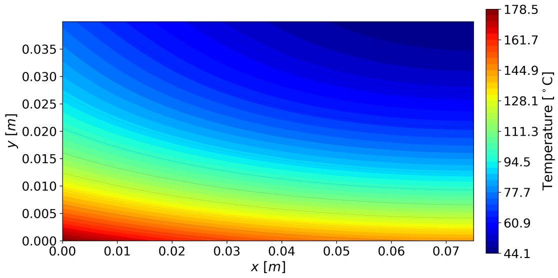
(c) Nodal absolute percentage error.

$q_b = 5709.4\ \mathrm{W/m^2}$
$q_l = 8189.8\ \mathrm{W/m^2}$
$t = 400\ \mathrm{s}$
$T_0 = 25\ °\mathrm{C}$

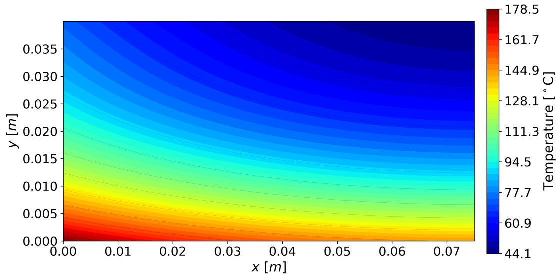
$\mathrm{RMSE} = 0.293\ °\mathrm{C}$
$\mathrm{MAPE} = 0.328\%$
$\mathrm{RMSE}_\theta = 1.343 \times 10^{-3}$

(d) Case parameters and error metrics.

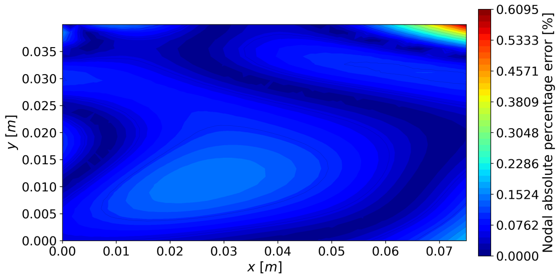
Figure 4.15: Dimensional contour comparison for case C08 at $t = 400$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

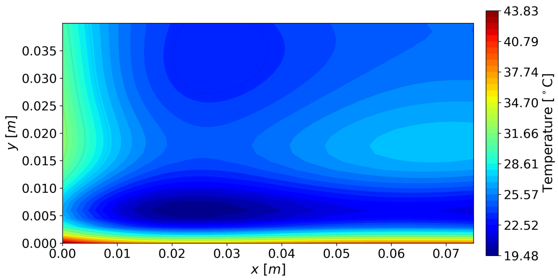


(c) Nodal absolute percentage error.

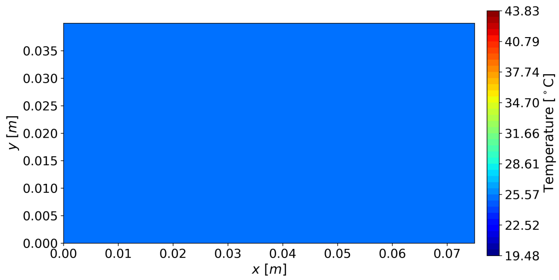
$q_b = 5709.4\ \mathrm{W/m^2}$
$q_l = 8189.8\ \mathrm{W/m^2}$
$t = 600\ \mathrm{s}$
$T_0 = 25\ °\mathrm{C}$
$\mathrm{RMSE} = 0.079\ °\mathrm{C}$
$\mathrm{MAPE} = 0.071\%$
$\mathrm{RMSE}_\theta = 3.627 \times 10^{-4}$

(d) Case parameters and error metrics.

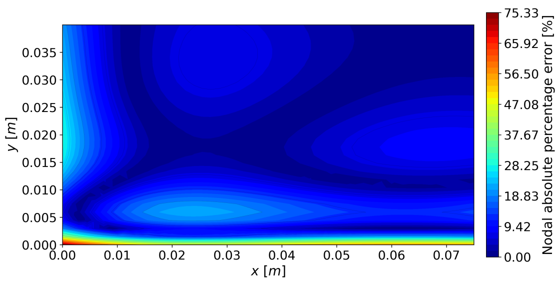
Figure 4.16: Dimensional contour comparison for case C08 at $t = 600$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

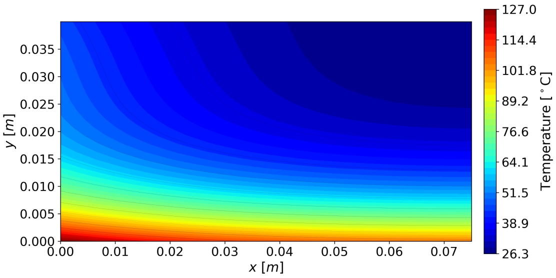


(c) Nodal absolute percentage error.

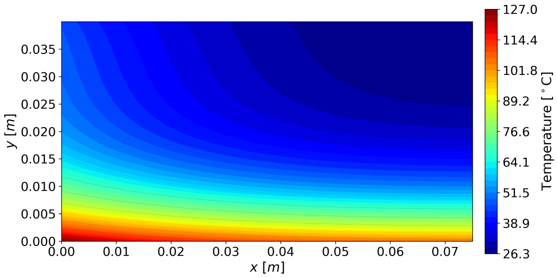
$q_b = 7543.5\ \mathrm{W/m^2}$
$q_l = 6731.1\ \mathrm{W/m^2}$
$t = 0\ \mathrm{s}$
$T_0 = 25\ °\mathrm{C}$
$\mathrm{RMSE} = 3.406\ °\mathrm{C}$
$\mathrm{MAPE} = 8.426\%$
$\mathrm{RMSE}_\theta = 1.563 \times 10^{-2}$

(d) Case parameters and error metrics.

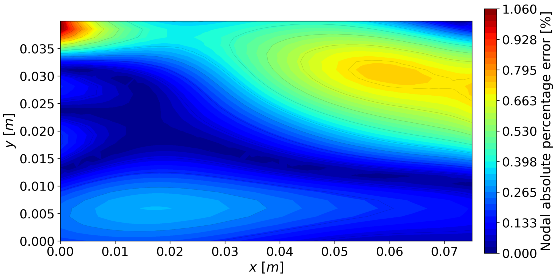
Figure 4.17: Dimensional contour comparison for case C13 at $t = 0$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

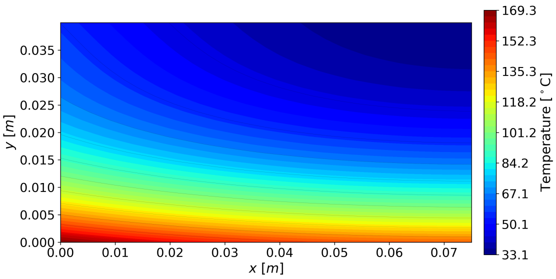


(c) Nodal absolute percentage error.

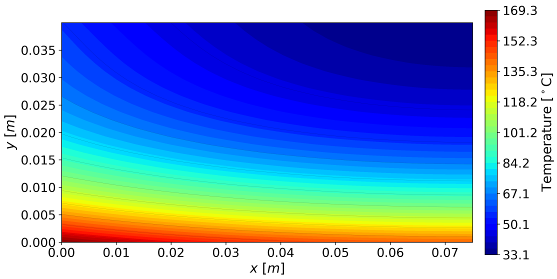
$q_b = 7543.5$ W/m^2
$q_l = 6731.1$ W/m^2
$t = 200$ s
$T_0 = 25$ °C
RMSE $= 0.143$ °C
MAPE $= 0.286\%$
RMSE$_\theta = 6.566 \times 10^{-4}$

(d) Case parameters and error metrics.

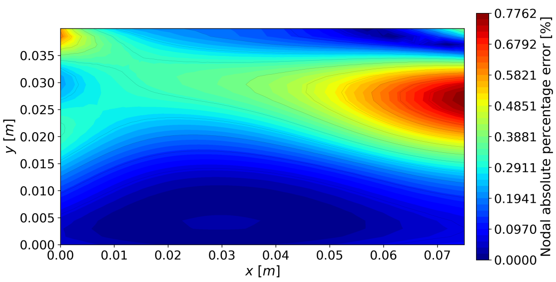
Figure 4.18: Dimensional contour comparison for case C13 at $t = 200$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

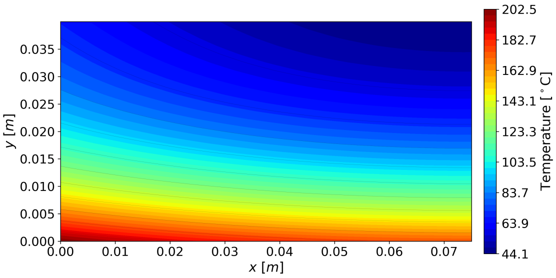


(c) Nodal absolute percentage error.

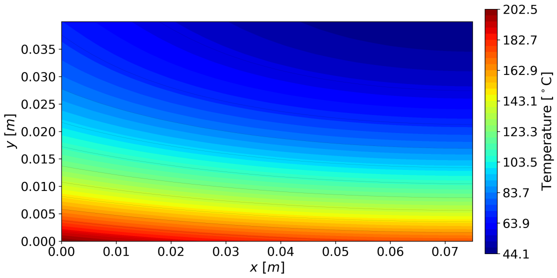
$q_b = 7543.5$ W/m^2
$q_l = 6731.1$ W/m^2
$t = 400$ s
$T_0 = 25$ °C
RMSE $= 0.143$ °C
MAPE $= 0.222\%$
RMSE$_\theta = 6.544 \times 10^{-4}$

(d) Case parameters and error metrics.

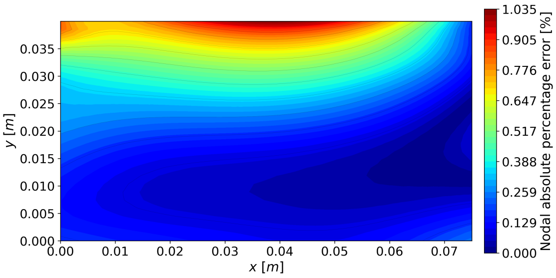
Figure 4.19: Dimensional contour comparison for case C13 at $t = 400$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

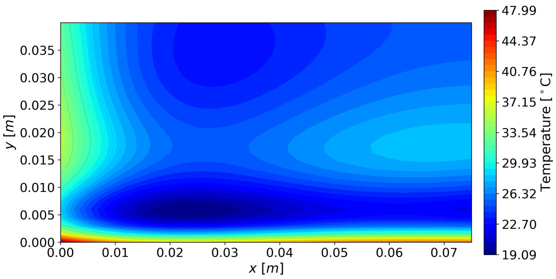


(c) Nodal absolute percentage error.

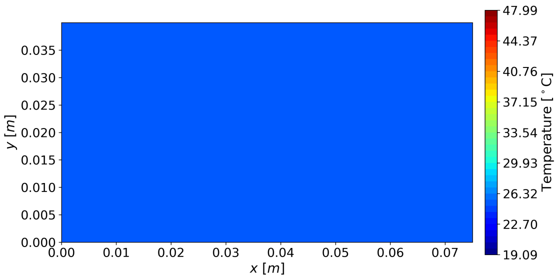
$q_b = 7543.5\ \mathrm{W/m^2}$
$q_l = 6731.1\ \mathrm{W/m^2}$
$t = 600\ \mathrm{s}$
$T_0 = 25\ ^\circ\mathrm{C}$
$\mathrm{RMSE} = 0.228\ ^\circ\mathrm{C}$
$\mathrm{MAPE} = 0.262\%$
$\mathrm{RMSE}_\theta = 1.046 \times 10^{-3}$

(d) Case parameters and error metrics.

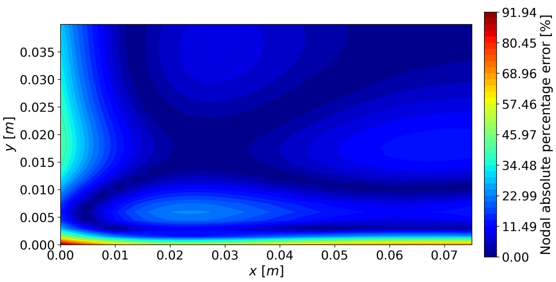
Figure 4.20: Dimensional contour comparison for case C13 at $t = 600$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

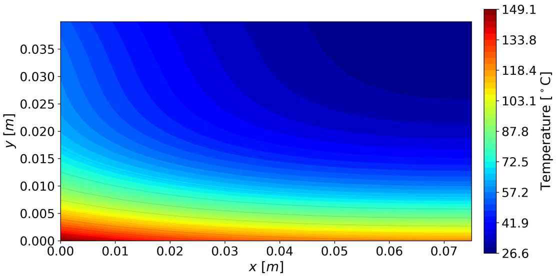


(c) Nodal absolute percentage error.

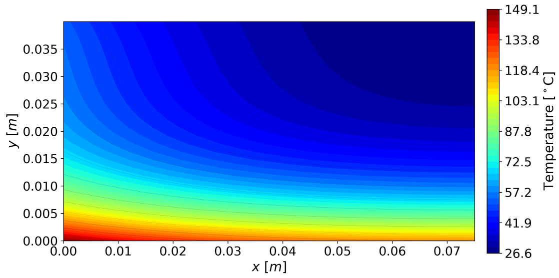
$q_b = 8867.8\ \mathrm{W/m^2}$
$q_l = 8787.3\ \mathrm{W/m^2}$
$t = 0\ \mathrm{s}$
$T_0 = 25\ ^\circ\mathrm{C}$
$\mathrm{RMSE} = 4.230\ ^\circ\mathrm{C}$
$\mathrm{MAPE} = 10.531\%$
$\mathrm{RMSE}_\theta = 1.941 \times 10^{-2}$

(d) Case parameters and error metrics.

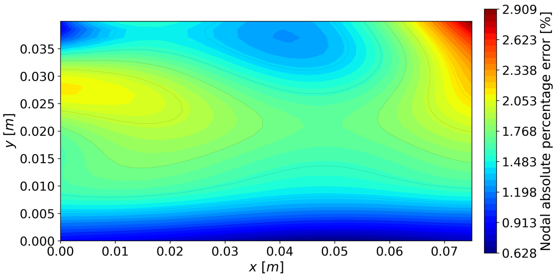
Figure 4.21: Dimensional contour comparison for case C11 at $t = 0$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

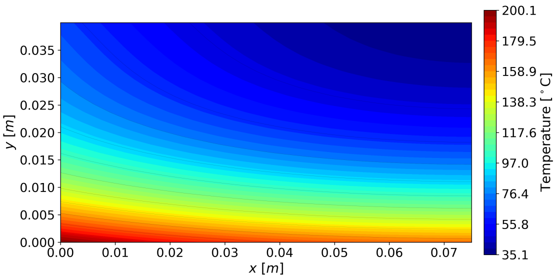


(c) Nodal absolute percentage error.

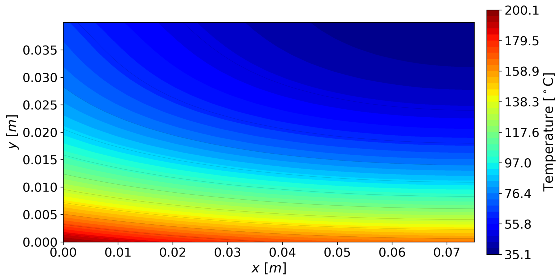
$q_b = 8867.8\ \mathrm{W/m^2}$
$q_l = 8787.3\ \mathrm{W/m^2}$
$t = 200$ s
$T_0 = 25$ °C
RMSE $= 0.839$ °C
MAPE $= 1.575\%$
$\mathrm{RMSE}_\theta = 3.848 \times 10^{-3}$

(d) Case parameters and error metrics.

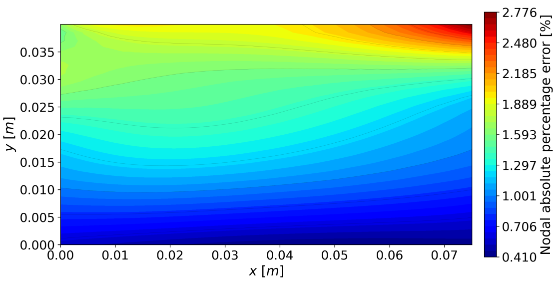
Figure 4.22: Dimensional contour comparison for case C11 at $t = 200$ s.



(a) PINN temperature field.



(b) Abaqus reference field.

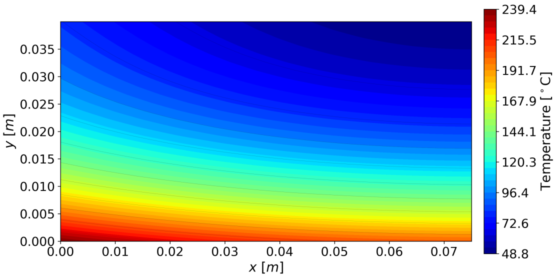


(c) Nodal absolute percentage error.

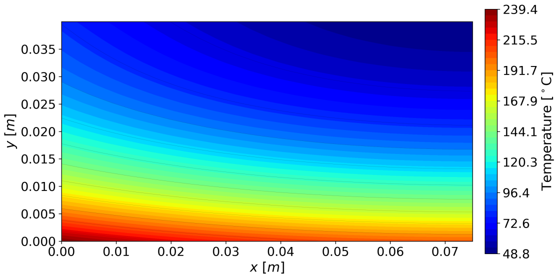
$q_b = 8867.8\ \mathrm{W/m^2}$
$q_l = 8787.3\ \mathrm{W/m^2}$
$t = 400$ s
$T_0 = 25$ °C
RMSE $= 0.907$ °C
MAPE $= 1.253\%$
$\mathrm{RMSE}_\theta = 4.163 \times 10^{-3}$

(d) Case parameters and error metrics.

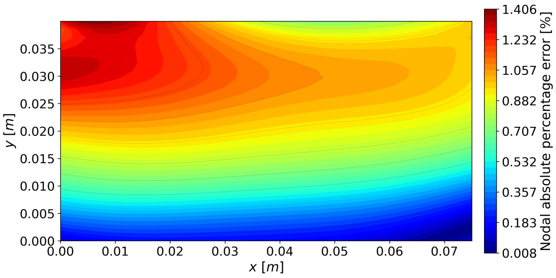
Figure 4.23: Dimensional contour comparison for case C11 at $t = 400$ s.



(a) PINN temperature field.



(b) Abaqus reference field.



(c) Nodal absolute percentage error.

$q_b = 8867.8 \text{ W/m}^2$
$q_l = 8787.3 \text{ W/m}^2$
$t = 600$ s
$T_0 = 25$ °C
RMSE $= 0.795$ °C
MAPE $= 0.806\%$
$\text{RMSE}_\theta = 3.650 \times 10^{-3}$

(d) Case parameters and error metrics.

Figure 4.24: Dimensional contour comparison for case C11 at $t = 600$ s.

4.2.3 Geometric sensitivity assessment

An additional geometric sensitivity assessment was performed to evaluate the response of the best-performing PINN configuration when applied to geometries slightly different from the one used during training. This analysis was motivated by the use of a non-dimensional formulation and was intended to assess whether the trained model remains accurate under small variations of the plate dimensions.

The best-performing model, `run_0484`, was kept fixed and was not retrained. New Abaqus reference solutions were generated using the same material properties, thermal BCs, IC, and Random25 heat-flux combinations used in the previous validation analysis. The horizontal dimension was kept fixed at $L_x = 0.075$ m, while the vertical dimension was increased from its original value $L_y = 0.040$ m up to $L_y = 0.042$ m in increments of 1%. Therefore, six geometries were evaluated:

$$L_y = 0.040(1+\delta_L), \qquad \delta_L \in \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}. \tag{4.1}$$

For each geometry, the same 25 Random25 heat-flux cases were solved in Abaqus. The geometric sensitivity assessment was performed at the final physical time, $t = 600$ s, in order to remain consistent with the validation criterion used to rank the PINN configurations in the sensitivity study. This resulted in $6 \times 25 = 150$ Abaqus simulations and 150 PINN–Abaqus comparisons for the geometric sensitivity analysis.

Since the model is formulated in terms of non-dimensional variables, the Abaqus temperature fields were converted to the non-dimensional temperature

$$\theta_{\mathrm{FEM}} = \frac{T_{\mathrm{FEM}} - T_0}{T_{\mathrm{ref}} - T_0} \tag{4.2}$$

and compared directly with the PINN prediction $\hat{\theta}$. Dimensional temperature errors were also computed as auxiliary metrics. However, the non-dimensional errors were used as the main indicators for this assessment because they are directly related to the formulation used by the PINN.

Table 4.4 summarises the case-averaged error metrics obtained for each geometric variation at $t = 600$ s. The overbar notation indicates averaging over the 25 Random25 heat-flux cases for each geometry.

Table 4.4: Geometric sensitivity of the best-performing PINN at $t = 600$ s.

ΔL_y [%]	L_y [m]	$\overline{\mathrm{MAE}}$ [°C]	$\overline{\mathrm{RMSE}}$ [°C]	$\overline{\mathrm{MAPE}}$ [%]	$\overline{\mathrm{MAE}}_\theta$ [-]	$\overline{\mathrm{RMSE}}_\theta$ [-]
0	0.0400	0.216	0.244	0.266	9.934×10^{-4}	1.122×10^{-3}
1	0.0404	0.474	0.512	0.577	2.174×10^{-3}	2.351×10^{-3}
2	0.0408	0.861	0.916	1.086	3.952×10^{-3}	4.205×10^{-3}
3	0.0412	1.249	1.322	1.601	5.733×10^{-3}	6.068×10^{-3}
4	0.0416	1.630	1.722	2.111	7.482×10^{-3}	7.904×10^{-3}
5	0.0420	2.006	2.118	2.617	9.207×10^{-3}	9.719×10^{-3}

The results show a systematic increase in the prediction error as the vertical dimension of the plate is increased. For the original geometry, the case-averaged MAPE was $\overline{\mathrm{MAPE}} = 0.266\%$, which is consistent with the validation performance previously obtained for the best-performing configuration. When the vertical dimension was increased by 5%, the case-averaged MAPE increased to 2.617%, while the dimensional RMSE increased from 0.244 °C to 2.118 °C. The same trend is observed in the non-dimensional metrics, with $\overline{\mathrm{RMSE}}_\theta$ increasing from 1.122×10^{-3} to 9.719×10^{-3}.

These results indicate that the trained PINN remains accurate for the original geometry and small geometric variations, but its error increases monotonically as the geometry departs from the training configuration. Therefore, although the non-dimensional formulation provides a consistent framework for comparing geometrically similar domains, the present model should not be interpreted as fully geometry-parametric. Future extensions should include geometric parameters explicitly as additional neural-network inputs if larger geometric variations are to be considered.

Overall, the geometric sensitivity assessment provides a complementary robustness test of the proposed methodology. The analysis shows that small variations of the vertical dimension can be evaluated using the trained non-dimensional PINN, but it also highlights the limitation of applying a model trained on a single geometry to domains with different aspect ratios.

4.3 Computational cost assessment

The computational cost of the proposed approach was assessed from two complementary perspectives. First, the GPU training time of the complete sensitivity study was analysed. Second, the CPU inference time of the best-performing PINN was compared with the execution time required by Abaqus for equivalent thermal simulations.

4.3.1 GPU training cost of the sensitivity study

The GPU training time of each PINN configuration was recorded during the sensitivity study. Table 4.5 summarises the training-time statistics obtained for the full set of 625 configurations. The reported statistics describe the distribution of GPU training times across all trained configurations, while the training time of the best-performing configuration is included for reference.

Table 4.5: Training-time summary for the sensitivity study.

Metric	Value
Number of configurations	625
Minimum training time [s]	12.714
Mean training time [s]	34.432
Median training time [s]	30.241
Maximum training time [s]	103.158
Standard deviation [s]	16.680
Training time of the best-performing configuration [s]	53.454
Total training time [h]	5.978

The total accumulated GPU training time for the complete sensitivity study was approximately 5.98 hours. The best-performing configuration required 53.454 s of GPU training time, which is higher than the mean and median training times of the full set but still well below the maximum observed value. This indicates that the selected configuration achieved the lowest case-averaged validation error, $\overline{\text{MAPE}}$, without requiring one of the most expensive training runs in the sensitivity study.

4.3.2 CPU inference-time comparison with Abaqus

A separate CPU-based timing study was performed to compare the inference cost of the trained PINN with the computational cost of Abaqus. The best-performing PINN obtained from the sensitivity study was used for this comparison. A total of 10 random heat-flux cases were generated using a fixed random seed of 1999, and each case was executed five times with both methods, resulting in 50 timing measurements per method.

Both approaches were evaluated on CPU using a single computational thread. For the PINN, PyTorch was forced to run on CPU with one thread. For Abaqus, the simulations were executed with `numCpus=1` and `numGPUs=0`. The comparison was carried out on a local machine equipped with a `12th Gen Intel(R) Core(TM) i7-12650H` processor.

Table 4.6 summarises the CPU timing results. The PINN required approximately 3.37 ms per prediction, whereas Abaqus required approximately 11.43 s per simulation. The mean speed-up, computed case by case, was approximately 3404. Therefore, for this CPU-based test, the trained PINN was about 3.4×10^3 times faster than Abaqus for obtaining the temperature-field prediction.

Table 4.6: CPU inference-time comparison between PINN and Abaqus.

Method	Mean [s]	Std. dev. [s]	Min. [s]	Median [s]	Max. [s]	CV [%]
PINN	0.003368	0.000404	0.002882	0.003268	0.004887	11.982
Abaqus	11.429845	1.672377	10.377388	10.545905	16.477610	14.632

These results highlight the advantage of the trained PINN as a surrogate model. Although the PINN requires an initial training stage, once trained it can provide temperature predictions almost instantaneously compared with a conventional finite element solve.

4.4 Comparative discussion

Table 4.7 positions the present work with respect to the PINN-based heat-transfer approach proposed by Zobeiry and Humfeld [38]. This reference provides a relevant methodological point of comparison because it also addresses transient heat transfer using PINNs and finite element validation. The main differences arise from the physical BCs, the parametrised variables, and the role assigned to finite element data during training.

Table 4.7: Comparison with the reference PINN-based approach [38].

Aspect	Reference study [38]	Present work
Physical problem	Transient heat transfer in composite parts heated by convection.	Transient heat conduction in a 2D orthotropic composite plate.
Thermal loading	Convective oven heating.	Prescribed heat fluxes with convective losses.
Parametric inputs	Convective coefficients h_1 and h_2.	Imposed heat fluxes q_b and q_l, introduced in the non-dimensional boundary-condition formulation.
Input–output map	$T = f(x, y, t, h_1, h_2)$.	$\hat{\theta} = \mathcal{N}_\omega(\chi, \psi, \tau, q_b, q_l)$.
Dimensional scope	1D and 2D.	2D
Geometry and material	Composite part; 2D case of $60\,\mathrm{mm} \times 20\,\mathrm{mm}$.	Carbon/polyamide 410 plate; $75\,\mathrm{mm} \times 40\,\mathrm{mm}$.
Role of FEM data	FEM results used for validation.	Abaqus finite-element solutions used as supervised training data and validation reference.
Loss weighting	Adaptive weights λ_i for PDE, BC, and IC residuals.	Fixed weights: $W_{\mathrm{data}} = 0.05$, $W_{\mathrm{PDE}} = 1.0$, $W_{\mathrm{BC}} = 1.0$, and $W_{\mathrm{IC}} = 1.0$.
Network architecture	PINN with theory-guided sinusoidal/exponential features.	Fully connected PINN with tanh activation.
Scaling strategy	Adaptive loss normalisation.	Non-dimensional variables and parameters.
Training strategy	Random collocation points; no FEM training data required.	Abaqus mini-batches plus random collocation points.
Validation set	Comparison with FEM solutions for selected 1D and 2D cases.	Comparison with Abaqus for 25 independent random heat-flux cases.
Main objective	Near real-time prediction for varying convective coefficients.	Fast Abaqus surrogate for varying imposed heat fluxes.

Both works share the same general motivation: reducing the need for repeated finite element simulations by training a neural model capable of rapidly evaluating transient thermal fields. However, they address this objective through different strategies. In Zobeiry and Humfeld [38], the PINN is mainly driven by the governing equation, BCs, and IC, while finite element results are used for validation. In the present work, Abaqus results are included during training, but with a reduced weight in the loss function. The supervised term therefore acts as a weak anchor within the heat-flux domain, while the PDE, BC, and IC residuals remain the dominant constraints.

This design choice is related to the loss-balancing problem discussed by Zobeiry and Humfeld [38]. In their work, adaptive normalisation is introduced to avoid one residual component dominating the optimisation. In this thesis, the same issue is addressed in a simpler way: the physics-based terms are assigned unit weights, whereas the data term is assigned a weight of 0.05. This fixed weighting scheme avoids introducing an additional adaptive variable into an already broad sensitivity study, allowing the analysis to focus on the selected hyperparameters: hidden layers, training epochs, learning rate, and collocation points. Thus, the model should be interpreted as a physics-informed surrogate weakly guided by finite element data.

The architectural comparison also highlights an important difference. Zobeiry and Humfeld [38] embed part of the expected thermal behaviour directly into the network through sinusoidal and exponential physics-informed features. In contrast, the present work uses a more direct fully connected architecture with (tanh) activation. This choice is consistent with the requirement discussed by Zobeiry and Humfeld [38]: for heat-transfer PINNs, the activation function must allow the computation of higher-order derivatives because the PDE residual includes second-order spatial derivatives. Unlike Rectified Linear Unit (ReLU), whose higher-order derivatives vanish, (tanh) is smooth and differentiable, making it compatible with the residual-based formulation adopted here. Thus, in this thesis, the physical structure is introduced mainly through non-dimensional variables and the residual terms in the loss function, rather than through a problem-specific feature layer.

Within this framework, the hidden-layer width was fixed at 128 neurons. This value is not presented as an optimal width, but as a modelling choice intended to provide a sufficiently expressive network while avoiding an additional hyperparameter sweep. Fixing the width helped isolate the effects of the parameters that were explicitly varied in the sensitivity study. At the same time, the results show that increasing model complexity does not automatically improve accuracy: deeper or more demanding configurations did not systematically produce lower validation errors. This confirms that PINN performance depends not only on network size, but also on optimisation stability, collocation sampling, variable scaling, and loss balance.

Overall, the comparison suggests that the present thesis follows a simpler and more controlled PINN formulation than that of Zobeiry and Humfeld [38]. While the reference study emphasises adaptive normalisation and physics-guided feature engineering, this work prioritises a non-dimensional heat-flux-parametric formulation and a fixed training design that enables a clear sensitivity analysis. Future work could combine both approaches by introducing adaptive loss weighting, physics-informed feature layers, or an explicit study of network width. However, these extensions would increase the dimensionality of the hyperparameter space and were therefore left outside the scope of the present study.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

The main objective of this work was to evaluate the capability of a parametric PINN to predict transient temperature fields in an orthotropic composite plate subjected to different heat-flux BCs. For this purpose, a hybrid FEM–PINN methodology was developed, in which Abaqus finite element simulations were used as reference data, while the governing transient heat-transfer equation, the BCs, and the IC were incorporated into the training loss.

The results obtained in this work show that a single trained parametric PINN can accurately approximate the thermal response of the composite plate within the prescribed range of imposed heat fluxes. By including the bottom and left heat fluxes as input parameters of the neural network, the model was able to predict temperature fields for loading conditions that were not part of the structured training set. This confirms the potential of the proposed formulation as a surrogate model for parametric transient heat-transfer problems.

The sensitivity study demonstrated that the performance of the PINN is strongly dependent on the selected training and architecture parameters. Among the factors analysed, the learning rate had the most significant influence on validation accuracy. Very high learning rates led to unstable optimisation, while very low learning rates slowed convergence within the prescribed number of epochs. The best results were obtained for intermediate learning rates, particularly $\alpha = 10^{-3}$. The number of hidden layers and the collocation level also affected the prediction accuracy, although increasing the network depth or the number of collocation points did not always lead to improved performance. These results highlight the importance of systematic calibration during any PINN training stage.

The best-performing configuration was obtained with five hidden layers, 128 neurons per hidden layer, $10,000$ training epochs, a learning rate of 10^{-3}, and the C4 collocation level, corresponding to $15,000$ interior collocation points, $3,750$ points per boundary, and $7,500$ initial-condition points. This model achieved a MAPE of 0.262% over the Random25 validation set. In dimensional terms, the corresponding MAE and RMSE were 0.216 °C and 0.240 °C, respectively. The maximum absolute percentage error was 1.458%, while the maximum absolute temperature error was 1.280 °C. These values indicate that the trained PINN reproduced the Abaqus reference temperature fields with high accuracy for independent heat-flux combinations within the analysed domain.

The spatial and temporal validation also confirmed that the PINN predictions were physically consistent with the expected thermal behaviour of the plate. The model correctly captured the increase in temperature near the heated boundaries and the lower temperature regions near the convective boundaries. The largest local errors were mainly concentrated near boundaries and corners, where strong thermal gradients and boundary-condition enforcement make the prediction task more demanding. Nevertheless, the overall error levels remained low, showing that the proposed model was able to capture the main features of the transient thermal response.

The geometric sensitivity assessment showed that the trained model remained accurate for the original geometry and for small variations of the plate height. However, the prediction error increased as the geometry departed from the configuration used during training. This indicates that, although the non-dimensional formulation provides some robustness for geometrically similar domains, the present model should be interpreted primarily as a heat-flux-parametric surrogate rather than as a fully geometry-parametric model.

From a computational point of view, the trained PINN provided a significant reduction in prediction time compared with conventional finite element simulations. Although the model requires an initial training stage, its inference time after training was approximately 0.00337 s per prediction, whereas Abaqus required approximately 11.43 s per simulation under equivalent CPU conditions. This corresponds to an average speed-up of approximately 3.4×10^3. Therefore, once trained, the PINN can provide almost instantaneous temperature-field predictions for new heat-flux combinations within the trained parameter range.

Overall, the central research question of this work can be answered positively. Within the assumptions, geometry, and heat-flux range considered, the parametric PINN was able to act as an accurate and computationally efficient surrogate model for transient thermal prediction in an orthotropic composite plate. The results demonstrate that PINNs are a promising tool for repeated thermal evaluations in composite materials, particularly when multiple loading conditions must be analysed and conventional finite element simulations become computationally expensive.

At the same time, the conclusions of this work should be understood within the limits of the adopted modelling framework. The material properties were considered constant, the analysis was restricted to a two-dimensional rectangular plate, and the validation was performed against finite element reference solutions rather than experimental measurements. In addition, the trained model should mainly be used for interpolation within the prescribed heat-flux range of 5000 to 9000 $\mathrm{W/m^2}$. Despite these limitations, the results provide a solid basis for the use of parametric PINNs as fast and accurate surrogate models for transient heat-transfer analysis in composite materials.

5.2 Future work

Future developments can extend the present work in several directions. A first improvement would be to include temperature-dependent material properties in the governing equation. This would allow the PINN to represent the non-linear thermal behaviour of thermoplastic composites more realistically, especially in cases where temperature variations are large enough to modify thermal conductivity, density, or specific heat capacity.

A second extension would be to apply the methodology to three-dimensional composite structures or multilayer laminates. This would make it possible to account for through-thickness heat transfer, stacking-sequence effects, and more representative composite geometries. In addition, more complex BCs could be considered, including spatially variable heat fluxes, time-dependent thermal loads, or different convection conditions.

Further work could also consider broader parametric spaces. In addition to the imposed heat fluxes, future PINN formulations could include case-dependent material properties, such as thermal conductivity, density, or specific heat capacity values associated with different composite systems or laminate configurations. Unlike the temperature-dependent properties mentioned above, these parameters would vary from one analysed case to another and would be treated as additional inputs to the neural network, in the same way that the imposed heat fluxes were considered in the present work. Other operating parameters, such as convection coefficients or ambient temperature, could also be incorporated to extend the applicability of the surrogate model for design exploration, optimisation, and rapid thermal assessment under different thermal conditions.

Another important direction is the use of adaptive training strategies. Adaptive collocation-point sampling, dynamic loss weighting, or hybrid optimisation algorithms could be investigated to improve the enforcement of BCs and reduce local errors near corners and heated boundaries. These strategies may also reduce the need for extensive sensitivity studies by improving training stability and convergence.

The proposed approach could also be compared with other surrogate modelling techniques, such as reduced-order models, DeepONet, Fourier Neural Operators, or purely data-driven neural networks. Such comparisons would help clarify the advantages and limitations of PINNs relative to alternative ML methods for parametric heat-transfer prediction.

Finally, experimental validation should be considered in future studies. Comparing PINN predictions with measured temperature fields from thermographic or sensor-based experiments would provide a stronger assessment of the method for practical composite-material applications. This would also help evaluate how the proposed surrogate model performs when uncertainty, measurement noise, and real manufacturing or service conditions are present.

References

[1] G. F. Abdelal, M. M. Abdalla, and Z. Gürdal. "Transient Heat Conduction of Variable Stiffness Composite Laminate". In: *Journal of Thermal Stresses* 33.2 (2010), pp. 121–134. DOI: 10.1080/01495730903454579. eprint: https://doi.org/10.1080/01495730903454579. URL: https://doi.org/10.1080/01495730903454579.

[2] Zulfiqar Ali et al. "Preparation, Properties and Mechanisms of Carbon Fiber/Polymer Composites for Thermal Management Applications". In: *Polymers* 13.1 (2021). ISSN: 2073-4360. DOI: 10.3390/polym13010169. URL: https://www.mdpi.com/2073-4360/13/1/169.

[3] Sina Amini Niaki et al. "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture". In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113959. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.113959. URL: https://www.sciencedirect.com/science/article/pii/S0045782521002966.

[4] Kasahun Niguse Asfew, Jan Ivens, and David Moens. "Temperature dependence of thermophysical properties of carbon/polyamide410 composite". In: *Functional Composite Materials* 3.1 (2022), p. 8.

[5] Simon Bard et al. "Influence of Fiber Volume Content on Thermal Conductivity in Transverse and Fiber Direction of Carbon Fiber-Reinforced Epoxy Laminates". In: *Materials* 12.7 (2019). ISSN: 1996-1944. DOI: 10.3390/ma12071084. URL: https://www.mdpi.com/1996-1944/12/7/1084.

[6] Brett Bowman et al. "Physics-Informed Neural Networks for the Heat Equation with Source Term under Various Boundary Conditions". In: *Algorithms* 16.9 (2023). ISSN: 1999-4893. DOI: 10.3390/a16090428. URL: https://www.mdpi.com/1999-4893/16/9/428.

[7] N. Burger et al. "Review of thermal conductivity in composites: Mechanisms, parameters and theory". In: *Progress in Polymer Science* 61 (2016). Topical Volume on Polymer Hybrid Materials, pp. 1–28. ISSN: 0079-6700. DOI: https://doi.org/10.1016/j.progpolymsci.2016.05.001. URL: https://www.sciencedirect.com/science/article/pii/S0079670016300156.

[8] Shengze Cai et al. "Physics-Informed Neural Networks for Heat Transfer Problems". In: *Journal of Heat Transfer* 143.6 (Apr. 2021), p. 060801. ISSN: 0022-1481. DOI: 10.1115/1.4050542. eprint: https://asmedigitalcollection.asme.org/heattransfer/article-pdf/143/6/060801/6688635/ht_143_06_060801.pdf. URL: https://doi.org/10.1115/1.4050542.

[9] Zhongliang Cao et al. “Temperature Field in the Heat Transfer Process of PEEK Thermoplastic Composite Fiber Placement”. In: *Materials* 13.19 (2020). ISSN: 1996-1944. DOI: 10.3390/ma13194417. URL: https://www.mdpi.com/1996-1944/13/19/4417.

[10] Xianqi Chen et al. “A deep neural network surrogate modeling benchmark for temperature field prediction of heat source layout”. In: *Science China Physics, Mechanics & Astronomy* 64.11 (2021), p. 1.

[11] Anastasios Danezis et al. “Heat transfer modelling of flashlamp heating for automated tape placement of thermoplastic composites”. In: *Composites Part A: Applied Science and Manufacturing* 145 (2021), p. 106381. ISSN: 1359-835X. DOI: https://doi.org/10.1016/j.compositesa.2021.106381. URL: https://www.sciencedirect.com/science/article/pii/S1359835X21001056.

[12] Xingyu Fu et al. “An finite element analysis surrogate model with boundary oriented graph embedding approach for rapid design”. In: *Journal of Computational Design and Engineering* 10.3 (2023), pp. 1026–1046.

[13] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[14] Myeong-Seok Go, Jae Hyuk Lim, and Seungchul Lee. “Physics-informed neural network-based surrogate model for a virtual thermal sensor with real-time simulation”. In: *International Journal of Heat and Mass Transfer* 214 (2023), p. 124392. ISSN: 0017-9310. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2023.124392. URL: https://www.sciencedirect.com/science/article/pii/S0017931023005380.

[15] J.S. Hesthaven and S. Ubbiali. “Non-intrusive reduced order modeling of nonlinear problems using neural networks”. In: *Journal of Computational Physics* 363 (2018), pp. 55–78. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.02.037. URL: https://www.sciencedirect.com/science/article/pii/S0021999118301190.

[16] E. Hosseini et al. “Single-track thermal analysis of laser powder bed fusion process: Parametric solution through physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 410 (2023), p. 116019. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2023.116019. URL: https://www.sciencedirect.com/science/article/pii/S0045782523001433.

[17] Marcin Kamiński. “Homogenization of transient heat transfer problems for some composite materials”. In: *International Journal of Engineering Science* 41.1 (2003), pp. 1–29. ISSN: 0020-7225. DOI: https://doi.org/10.1016/S0020-7225(02)00144-1. URL: https://www.sciencedirect.com/science/article/pii/S0020722502001441.

[18] Seid Koric and Diab W. Abueidda. “Data-driven and physics-informed deep learning operators for solution of heat conduction equation with parametric heat source”. In: *International Journal of Heat and Mass Transfer* 203 (2023), p. 123809. ISSN: 0017-9310. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2022.123809. URL: https://www.sciencedirect.com/science/article/pii/S0017931022012777.

[19] Aditi Krishnapriyan et al. “Characterizing possible failure modes in physics-informed neural networks”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 26548–26560. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af7 Paper.pdf.

[20] Jakub Kudela and Radomil Matousek. “Recent advances and applications of surrogate models for finite element method computations: a review: J. Kudela, R. Matousek”. In: *Soft Computing* 26.24 (2022), pp. 13709–13733.

[21] I.E. Lagaris, A. Likas, and D.I. Fotiadis. “Artificial neural networks for solving ordinary and partial differential equations”. In: *IEEE Transactions on Neural Networks* 9.5 (1998), pp. 987–1000. DOI: 10.1109/72.712178.

[22] Lu Lu et al. “DeepXDE: A Deep Learning Library for Solving Differential Equations”. In: *SIAM Review* 63.1 (2021), pp. 208–228. DOI: 10.1137/19M1274067. eprint: https://doi.org/10.1137/19M1274067. URL: https://doi.org/10.1137/19M1274067.

[23] Seyedalborz Manavi, Thomas Becker, and Ehsan Fattahi. “Enhanced surrogate modelling of heat conduction problems using physics-informed neural network framework”. In: *International Communications in Heat and Mass Transfer* 142 (2023), p. 106662. ISSN: 0735-1933. DOI: https://doi.org/10.1016/j.icheatmasstransfer.2023.106662. URL: https://www.sciencedirect.com/science/article/pii/S0735193323000519.

[24] Wiem Nasri et al. “Thermal Study of Carbon-Fiber-Reinforced Polymer Composites Using Multiscale Modeling”. In: *Materials* 16.22 (2023). ISSN: 1996-1944. DOI: 10.3390/ma16227233. URL: https://www.mdpi.com/1996-1944/16/22/7233.

[25] MW Pilling et al. “The thermal conductivity of carbon fibre-reinforced composites”. In: *Journal of materials science* 14.6 (1979), pp. 1326–1338.

[26] 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”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[27] R Rolfes and U Hammerschmidt. “Transverse thermal conductivity of CFRP laminates: A numerical and experimental validation of approximation formulae”. In: *Composites Science and Technology* 54.1 (1995), pp. 45–54.

[28] Thomas Schneider et al. “Machine learning based surrogate models for the thermal behavior of multi-plate clutches”. In: *Applied System Innovation* 5.5 (2022), p. 97.

[29] Fazil O. Sonmez and H. Thomas Hahn. “Modeling of Heat Transfer and Crystallization in Thermoplastic Composite Tape Placement Process”. In: *Journal of Thermoplastic Composite Materials* 10.3 (1997), pp. 198–240. DOI: 10.1177/089270579701000301. eprint: https://doi.org/10.1177/089270579701000301. URL: https://doi.org/10.1177/089270579701000301.

[30] Omid Aghababaei Tafreshi et al. “Heat transfer analysis of automated fiber placement of thermoplastic composites using a hot gas torch”. In: *Advanced Manufacturing: Polymer & Composites Science* 5.4 (2019), pp. 206–223.

[31] Mingxuan Tian et al. “Physics-informed machine learning-based real-time long-horizon temperature fields prediction in metallic additive manufacturing”. In: *Communications Engineering* 4.1 (2025), p. 168.

[32] Poojitha Vurtur Badarinath, Maria Chierichetti, and Fatemeh Davoudi Kakhki. “A machine learning approach as a surrogate for a finite element analysis: Status of research and application to one dimensional systems”. In: *Sensors* 21.5 (2021), p. 1654.

[33] Sifan Wang, Yujun Teng, and Paris Perdikaris. “Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks”. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3055–A3081. DOI: 10.1137/20M1318043. eprint: https://doi.org/10.1137/20M1318043. URL: https://doi.org/10.1137/20M1318043.

[34] Sifan Wang et al. “An Expert’s Guide to Training Physics-informed Neural Networks”. In: *ArXiv* abs/2308.08468 (2023). URL: https://api.semanticscholar.org/CorpusID:260925531.

[35] Tobias Würth et al. “Physics-informed neural networks for data-free surrogate modelling and engineering optimization – An example from composite manufacturing”. In: *Materials & Design* 231 (2023), p. 112034. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2023.112034. URL: https://www.sciencedirect.com/science/article/pii/S0264127523004495.

[36] Zebin Xing, Heng Cheng, and Jing Cheng. “Deep Learning Method Based on Physics-Informed Neural Network for 3D Anisotropic Steady-State Heat Conduction Problems”. In: *Mathematics* 11.19 (2023). ISSN: 2227-7390. DOI: 10.3390/math11194049. URL: https://www.mdpi.com/2227-7390/11/19/4049.

[37] Xiaoyu Zhao et al. “A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout”. In: *Structural and Multidisciplinary Optimization* 64.4 (2021), pp. 2287–2306.

[38] Navid Zobeiry and Keith D. Humfeld. “A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications”. In: *Engineering Applications of Artificial Intelligence* 101 (2021), p. 104232. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2021.104232. URL: https://www.sciencedirect.com/science/article/pii/S0952197621000798.

Appendix A

Derivation of the Non-dimensional Heat-transfer Formulation

This appendix presents the derivation of the nondimensional governing equation and boundary-condition residuals used in the PINN formulation. The dimensional transient heat-transfer equation for an orthotropic plate is

$$\rho c_p \frac{\partial T}{\partial t} = k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2}, \qquad 0 < x < L_x, \quad 0 < y < L_y, \quad t > 0. \tag{A.1}$$

The nondimensional variables are defined as

$$\chi = \frac{x}{L_{\text{ref}}}, \tag{A.2}$$

$$\psi = \frac{y}{H_{\text{ref}}}, \tag{A.3}$$

$$\tau = \frac{k_{\text{ref}}}{\rho c_p L_{\text{ref}}^2} t, \tag{A.4}$$

and

$$\theta = \frac{T - T_0}{\Delta T_{\text{ref}}}, \qquad \Delta T_{\text{ref}} = T_{\text{ref}} - T_0. \tag{A.5}$$

Therefore, the dimensional temperature can be written as

$$T = T_0 + \Delta T_{\text{ref}} \theta. \tag{A.6}$$

The imposed heat fluxes are normalised as

$$\bar{q}_b = \frac{q_b}{q_{\text{ref}}}, \tag{A.7}$$

$$\bar{q}_l = \frac{q_l}{q_{\text{ref}}}. \tag{A.8}$$

For the present formulation, $L_{\text{ref}} = L_x$, $H_{\text{ref}} = L_y$, and $k_{\text{ref}} = k_x$.

A.1 Transformation of derivatives

Using Eq. (A.6), the dimensional temperature can be differentiated with respect to the dimensional time t. Since the derivative is taken at fixed x and y, Eqs. (A.2) and (A.3) imply that χ and ψ remain constant. Hence, the time dependence of θ occurs only through the nondimensional time τ. Applying the chain rule gives

$$\frac{\partial T}{\partial t} = \Delta T_{\text{ref}} \frac{\partial \theta}{\partial \tau}\frac{\partial \tau}{\partial t}. \tag{A.9}$$

From Eq. (A.4),

$$\frac{\partial \tau}{\partial t} = \frac{k_{\text{ref}}}{\rho c_p L_{\text{ref}}^2}. \tag{A.10}$$

Therefore,

$$\frac{\partial T}{\partial t} = \Delta T_{\text{ref}} \frac{k_{\text{ref}}}{\rho c_p L_{\text{ref}}^2}\frac{\partial \theta}{\partial \tau}. \tag{A.11}$$

The spatial derivatives are obtained in the same way. When differentiating with respect to x, the variables y and t are held fixed. Therefore, from Eqs. (A.3) and (A.4), ψ and τ remain constant, and the dependence of θ on x occurs only through χ. Using Eq. (A.2), the chain rule gives

$$\frac{\partial T}{\partial x} = \Delta T_{\text{ref}} \frac{\partial \theta}{\partial \chi}\frac{\partial \chi}{\partial x} = \Delta T_{\text{ref}} \frac{1}{L_{\text{ref}}}\frac{\partial \theta}{\partial \chi}. \tag{A.12}$$

Applying the same transformation once more,

$$\frac{\partial^2 T}{\partial x^2} = \Delta T_{\text{ref}} \frac{1}{L_{\text{ref}}}\frac{\partial}{\partial x}\left(\frac{\partial \theta}{\partial \chi}\right) = \Delta T_{\text{ref}} \frac{1}{L_{\text{ref}}^2}\frac{\partial^2 \theta}{\partial \chi^2}. \tag{A.13}$$

Similarly, when differentiating with respect to y, the variables x and t are held fixed. Therefore, from Eqs. (A.2) and (A.4), χ and τ remain constant, and the dependence of θ on y occurs only through ψ. Using Eq. (A.3),

$$\frac{\partial T}{\partial y} = \Delta T_{\text{ref}} \frac{\partial \theta}{\partial \psi}\frac{\partial \psi}{\partial y} = \Delta T_{\text{ref}} \frac{1}{H_{\text{ref}}}\frac{\partial \theta}{\partial \psi}. \tag{A.14}$$

Applying the same transformation once more,

$$\frac{\partial^2 T}{\partial y^2} = \Delta T_{\text{ref}} \frac{1}{H_{\text{ref}}}\frac{\partial}{\partial y}\left(\frac{\partial \theta}{\partial \psi}\right) = \Delta T_{\text{ref}} \frac{1}{H_{\text{ref}}^2}\frac{\partial^2 \theta}{\partial \psi^2}. \tag{A.15}$$

A.2 Nondimensional governing equation

Substituting Eqs. (A.11), (A.13), and (A.15) into the dimensional heat equation, Eq. (A.1), gives

$$\rho c_p \left(\Delta T_{\text{ref}} \frac{k_{\text{ref}}}{\rho c_p L_{\text{ref}}^2} \frac{\partial \theta}{\partial \tau} \right) = k_x \left(\Delta T_{\text{ref}} \frac{1}{L_{\text{ref}}^2} \frac{\partial^2 \theta}{\partial \chi^2} \right) + k_y \left(\Delta T_{\text{ref}} \frac{1}{H_{\text{ref}}^2} \frac{\partial^2 \theta}{\partial \psi^2} \right). \tag{A.16}$$

After cancelling the common factor ΔT_{ref}, the equation becomes

$$\frac{k_{\text{ref}}}{L_{\text{ref}}^2} \frac{\partial \theta}{\partial \tau} = \frac{k_x}{L_{\text{ref}}^2} \frac{\partial^2 \theta}{\partial \chi^2} + \frac{k_y}{H_{\text{ref}}^2} \frac{\partial^2 \theta}{\partial \psi^2}. \tag{A.17}$$

Dividing by $k_{\text{ref}}/L_{\text{ref}}^2$, the general nondimensional form is obtained as

$$\frac{\partial \theta}{\partial \tau} = \frac{k_x}{k_{\text{ref}}} \frac{\partial^2 \theta}{\partial \chi^2} + \frac{k_y}{k_{\text{ref}}} \left(\frac{L_{\text{ref}}}{H_{\text{ref}}} \right)^2 \frac{\partial^2 \theta}{\partial \psi^2}. \tag{A.18}$$

For the present formulation, $k_{\text{ref}} = k_x$, so the coefficient of the χ-diffusion term is equal to one. Therefore,

$$\frac{\partial \theta}{\partial \tau} = \frac{\partial^2 \theta}{\partial \chi^2} + \beta_y \frac{\partial^2 \theta}{\partial \psi^2}, \tag{A.19}$$

where

$$\beta_y = \frac{k_y}{k_{\text{ref}}} \left(\frac{L_{\text{ref}}}{H_{\text{ref}}} \right)^2. \tag{A.20}$$

Thus, the corresponding PDE residual used in the PINN formulation is

$$r_{\text{PDE}} = \frac{\partial \hat{\theta}}{\partial \tau} - \frac{\partial^2 \hat{\theta}}{\partial \chi^2} - \beta_y \frac{\partial^2 \hat{\theta}}{\partial \psi^2}. \tag{A.21}$$

A.3 Initial condition

The dimensional initial condition is

$$T(x, y, 0) = T_0. \tag{A.22}$$

At $t = 0$, Eq. (A.4) gives $\tau = 0$. Using the dimensionless temperature definition,

$$\theta = \frac{T - T_0}{\Delta T_{\text{ref}}}, \tag{A.23}$$

the initial condition becomes

$$\theta(\chi, \psi, 0, \bar{q}_b, \bar{q}_l) = \frac{T_0 - T_0}{\Delta T_{\text{ref}}} = 0. \tag{A.24}$$

Thus, the corresponding initial-condition residual used in the PINN formulation is

$$r_{\mathrm{IC}} = \hat{\theta}(\chi, \psi, 0, \bar{q}_b, \bar{q}_l). \tag{A.25}$$

A.4 Bottom heat-flux boundary

On the bottom boundary, $\psi = 0$, the imposed heat flux is taken as positive when entering the plate. The dimensional boundary condition is

$$-k_y \frac{\partial T}{\partial y} = q_b, \qquad y = 0. \tag{A.26}$$

Substituting Eq. (A.14) into Eq. (A.26), and using $q_b = q_{\mathrm{ref}} \bar{q}_b$, gives

$$-k_y \frac{\Delta T_{\mathrm{ref}}}{H_{\mathrm{ref}}} \frac{\partial \theta}{\partial \psi} = q_{\mathrm{ref}} \bar{q}_b. \tag{A.27}$$

Dividing by the reference conductive scale $k_{\mathrm{ref}} \Delta T_{\mathrm{ref}} / H_{\mathrm{ref}}$, the dimensionless bottom heat-flux condition becomes

$$-\frac{k_y}{k_{\mathrm{ref}}} \frac{\partial \theta}{\partial \psi} - \bar{q}_b \frac{q_{\mathrm{ref}} H_{\mathrm{ref}}}{k_{\mathrm{ref}} \Delta T_{\mathrm{ref}}} = 0. \tag{A.28}$$

Thus, the bottom-boundary residual used in the PINN formulation is

$$r_{\mathrm{bottom}} = -\frac{k_y}{k_{\mathrm{ref}}} \frac{\partial \hat{\theta}}{\partial \psi} - \bar{q}_b \frac{q_{\mathrm{ref}} H_{\mathrm{ref}}}{k_{\mathrm{ref}} \Delta T_{\mathrm{ref}}}. \tag{A.29}$$

A.5 Left heat-flux boundary

On the left boundary, $\chi = 0$, the imposed heat flux is also taken as positive when entering the plate. The dimensional boundary condition is

$$-k_x \frac{\partial T}{\partial x} = q_l, \qquad x = 0. \tag{A.30}$$

Substituting Eq. (A.12) into Eq. (A.30), and using $q_l = q_{\mathrm{ref}} \bar{q}_l$, gives

$$-k_x \frac{\Delta T_{\mathrm{ref}}}{L_{\mathrm{ref}}} \frac{\partial \theta}{\partial \chi} = q_{\mathrm{ref}} \bar{q}_l. \tag{A.31}$$

Dividing by the reference conductive scale $k_{\mathrm{ref}} \Delta T_{\mathrm{ref}} / L_{\mathrm{ref}}$, the dimensionless left heat-flux condition becomes

$$-\frac{k_x}{k_{\mathrm{ref}}} \frac{\partial \theta}{\partial \chi} - \bar{q}_l \frac{q_{\mathrm{ref}} L_{\mathrm{ref}}}{k_{\mathrm{ref}} \Delta T_{\mathrm{ref}}} = 0. \tag{A.32}$$

Thus, the left-boundary residual used in the PINN formulation is

$$r_{\mathrm{left}} = -\frac{k_x}{k_{\mathrm{ref}}} \frac{\partial \hat{\theta}}{\partial \chi} - \bar{q}_l \frac{q_{\mathrm{ref}} L_{\mathrm{ref}}}{k_{\mathrm{ref}} \Delta T_{\mathrm{ref}}}. \tag{A.33}$$

A.6 Top convective boundary

On the top boundary, $\psi = 1$, convective heat loss is imposed as

$$-k_y \frac{\partial T}{\partial y} = h_{\text{top}} \left(T - T_{\infty,\text{top}} \right), \qquad y = H_{\text{ref}}. \tag{A.34}$$

Substituting Eq. (A.14) and using $T = T_0 + \Delta T_{\text{ref}} \theta$, gives

$$-k_y \frac{\Delta T_{\text{ref}}}{H_{\text{ref}}} \frac{\partial \theta}{\partial \psi} = h_{\text{top}} \Delta T_{\text{ref}} \left(\theta - \theta_{\infty,\text{top}} \right), \tag{A.35}$$

where the dimensionless ambient temperature is

$$\theta_{\infty,\text{top}} = \frac{T_{\infty,\text{top}} - T_0}{\Delta T_{\text{ref}}}. \tag{A.36}$$

Dividing by $k_{\text{ref}} \Delta T_{\text{ref}} / H_{\text{ref}}$, the dimensionless top convective condition becomes

$$-\frac{k_y}{k_{\text{ref}}} \frac{\partial \theta}{\partial \psi} - \text{Bi}_{\text{top}} \left(\theta - \theta_{\infty,\text{top}} \right) = 0, \tag{A.37}$$

with

$$\text{Bi}_{\text{top}} = \frac{h_{\text{top}} H_{\text{ref}}}{k_{\text{ref}}}. \tag{A.38}$$

Equivalently, multiplying Eq. (A.37) by -1, the residual used in the PINN formulation is

$$r_{\text{top}} = \frac{k_y}{k_{\text{ref}}} \frac{\partial \hat{\theta}}{\partial \psi} + \text{Bi}_{\text{top}} \left(\hat{\theta} - \theta_{\infty,\text{top}} \right). \tag{A.39}$$

A.7 Right convective boundary

On the right boundary, $\chi = 1$, convective heat loss is imposed as

$$-k_x \frac{\partial T}{\partial x} = h_{\text{right}} \left(T - T_{\infty,\text{right}} \right), \qquad x = L_{\text{ref}}. \tag{A.40}$$

Substituting Eq. (A.12) and using $T = T_0 + \Delta T_{\text{ref}} \theta$, gives

$$-k_x \frac{\Delta T_{\text{ref}}}{L_{\text{ref}}} \frac{\partial \theta}{\partial \chi} = h_{\text{right}} \Delta T_{\text{ref}} \left(\theta - \theta_{\infty,\text{right}} \right), \tag{A.41}$$

where the dimensionless ambient temperature is

$$\theta_{\infty,\text{right}} = \frac{T_{\infty,\text{right}} - T_0}{\Delta T_{\text{ref}}}. \tag{A.42}$$

Dividing by $k_{\text{ref}} \Delta T_{\text{ref}} / L_{\text{ref}}$, the dimensionless right convective condition becomes

$$-\frac{k_x}{k_{\text{ref}}} \frac{\partial \theta}{\partial \chi} - \text{Bi}_{\text{right}} \left(\theta - \theta_{\infty,\text{right}} \right) = 0, \tag{A.43}$$

with

$$\mathrm{Bi}_{\mathrm{right}} = \frac{h_{\mathrm{right}} L_{\mathrm{ref}}}{k_{\mathrm{ref}}}. \tag{A.44}$$

Equivalently, multiplying Eq. (A.43) by -1, the residual used in the PINN formulation is

$$r_{\mathrm{right}} = \frac{k_x}{k_{\mathrm{ref}}} \frac{\partial \hat{\theta}}{\partial \chi} + \mathrm{Bi}_{\mathrm{right}} \left(\hat{\theta} - \theta_{\infty,\mathrm{right}} \right). \tag{A.45}$$

Appendix B

Computational run identifiers

This appendix reports the internal computational identifiers used to trace the trained PINN models discussed in Chapter 4. These identifiers were used only for data management, reproducibility, and post-processing. They are not used as the primary notation in the main text, where each model is described in terms of its hyperparameter values.

Table B.1: Internal identifiers for the representative PINN configurations discussed in Chapter 4.

Case description	Run ID	Architecture code	n_h	Width	N_{ep}	α
High learning-rate case	`run_0479`	`H5_W128`	5	128	10 000	10^{-2}
Best-performing model	`run_0484`	`H5_W128`	5	128	10 000	10^{-3}
Low learning-rate case	`run_0499`	`H5_W128`	5	128	10 000	10^{-6}

The architecture code Hn_h_Ww denotes a fully connected neural network with n_h hidden layers and w neurons per hidden layer. For example, `H5_W128` corresponds to a network with five hidden layers and 128 neurons per hidden layer.