



Enhanced surrogate modelling of heat conduction problems using physics-informed neural network framework

Seyedalborz Manavi, Thomas Becker, Ehsan Fattahi*

Chair of Brewing and Beverage Technology, Technical University of Munich, Freising, Germany



ARTICLE INFO

Keywords:

Machine learning
Hybrid enforcement
Hard constraint
Thermal simulation
Irregular geometries

ABSTRACT

Solving partial differential equations (PDEs) using deep-learning techniques provides opportunities for surrogate models that require no labelled data, e.g., CFD results, from the domain interior other than the boundary and initial conditions. We propose a new ansatz of the solution incorporated with a physics-informed neural network (PINN) for solving PDEs to impose the boundary conditions (BCs) with hard constraints. This ansatz comprises three subnetworks: a boundary function, a distance function, and a deep neural network (DNN). The new model performance is assessed thoroughly in terms of convergence speed and accuracy. To this end, we apply the PINN models to conduction heat transfer problems with different geometries and BCs. The results of 1D, 2D and 3D problems are compared with conventional numerical methods and analytical results. The results reveal that the neural networks (NNs) model with the proposed ansatz outperforms counterpart PINN models in the literature and leads to faster convergence with better accuracy, especially for higher dimensions, i.e., three-dimensional case studies.

1. Introduction

Partial differential equations (PDEs) play a crucial role in several engineering branches. The behaviour of many physical phenomena, such as fluid flow, heat transfer, wave propagation and elasticity, are translated into mathematical formulations using PDEs [1,2]. For centuries, there has been an ever-increasing demand for efficient and accurate solutions to these equations to understand and control these phenomena better. Since analytical solutions to most of them are impractical, numerical approaches with different discretization methods have been developed to approximate solutions to these equations. Despite tremendous progress, these conventional numerical solvers are computationally cost-ineffective, especially in complex systems with multiscale and multiphysics properties.

Recent advancements in artificial intelligence, and especially its subset, deep learning, have revolutionised various disciplines in computer vision [3], language processing [4] and bioinformatics [5]. Deep learning has also been adopted across several applications in computational mechanics by learning the dynamical system models from data [6] and modified differential equations analysis [7]. Consequently, the data-driven approach to PDE problems emerged as a new framework that has attracted considerable scholarly attention, specifically in

computational fluid dynamics [8]. For example, this methodology was employed to improve the accuracy of the turbulent RANS (Reynolds-averaged Navier-Stokes) model [9], model turbulent flow in complex systems [10] and reconstruct the entire fluid velocity field from a set of reduced parameters by a generative fluid convolutional neural network [11]. More recently, operator learning with deep neural networks (DNNs) has been proposed in which the PDE operators are learnt from data [12–14]. One branch of studies has practised data-driven technique in heat transfer problems to learn the physics of heat transfer mechanisms using the convolutional neural network [15] and estimate the convective heat transfer coefficient using the random forest technique, which was trained with numerical simulation data [16]. Lin et al. [17] built a supervised deep-learning predictor using a convolutional autoencoder neural network to infer conductive heat transfer topologies instantaneously. Wang et al. [18] applied deep learning to temperature field prediction in porous media with variable boundary conditions (BCs).

The common challenges of pure data-driven-based deep learning tasks are high reliance on massive data sets, the computational costs for training the model, model overfitting traps and poor generalisation ability. These issues have been addressed to some extent by incorporating the wealth of scientific knowledge into data-driven approaches, also known as theory-guided data science [19]. A variant of theory-

* Corresponding author.

E-mail address: ehsan.fattahi@tum.de (E. Fattahi).

Nomenclature		Greek symbols
x, y	Spatial coordinate	θ Neural network parameters
T	Non-dimensional Temperature ($\frac{T - T_{min}}{T_{max} - T_{min}}$)	Ω Calculation domain
R^d	Continuous domain	Γ Boundary domain
G	Neural network of boundary values	λ Relative weighting coefficient
D	Neural network of distance values	
N^T	Deep neural network for temperature	
$\mathcal{N}[\cdot]$	Nonlinear operator	
r	Radius	
q	Heat flux	
N	node numbers	
$loss$	Loss values of neural network	
C_1, C_2	Coefficients of analytical solution	
d	Minimum distances to the boundaries	
B	Boundary condition operator	
Subscripts		Subscripts
		n Total number of nodes
		b Boundary nodes
		i The index of nodes
		c Cold
		h Hot
		D Dirichlet
		Ne Neumann
Abbreviation		Abbreviation
		BC boundary condition

guided deep learning has been proposed by Sun et al. [20], forcing the non-trainable parameters of a recurrent neural network to mimic wave propagation. For thermodynamics problems, a new scheme based on flash calculation was developed by Zhang et al. [21], known as a thermodynamics-informed neural network (TINN). Physics-informed neural networks (PINNs), established by Raissi et al. [22], introduce a new paradigm of surrogate models for solving PDEs and data-driven discovery of PDEs that covers a wide range of engineering and scientific applications. In this approach, the governing equations are embedded into the neural network by penalising the loss function, resulting in a deep-learning framework relying on physical laws and less data that leads to a better generalisation ability. Different versions of PINNs have been developed since, aiming to tackle the current challenges by decomposing the domain into spatiotemporal sub-domains, implementing adaptive activation functions and normalising the scales of loss terms [23,24]. Shukla et al. [25] proposed an enhanced version of the PINN by implementing adaptive activation functions, which have been integrated with acoustic wave correlation for the non-destructive quantification of surface-breaking cracks. Many researchers focused on discovering the applicability of PINN models to different PDE problems. Eivazi et al. [26] investigated the capability of PINN for the solution estimation of RANS equations without further model assumptions required in the conventional numerical approach. Some studies were more dedicated to heat transfer problems. For instance, Cai et al. [27] used a PINN model to solve forced and mixed convection heat transfer problems with a limited number of temperature measurements and partly known BCs to infer other fluid properties of the entire domain. Niaki et al. [28] proposed a PINN model based on sequential training to approximate the solution of a one-dimensional (1D) unsteady-state-coupled conduction-based heat transfer with an exothermic process problem in a disjointed solid domain using the Heaviside function to apply the domain discontinuities in NNs explicitly. He et al. [29] presented a data-driven framework based on PINNs to accomplish the direct analysis and parameter inversion of heat conduction problems.

It must be mentioned that PINNs are not the first framework aimed at solving forward problems with neural networks. The history of solving PDEs with artificial neural networks dates back to 1997 when Lagaris et al. [30] analytically calculated the network derivatives with respect to its input variables. However, this approach has been replaced with a modern technique of automatic differentiation (AD) [31] in the PINN model, which is free of approximation error, leading to an accurate estimation of derivatives at the machine precision level. Since the uniqueness of PDE solutions is subject to the specified boundary and initial conditions, this is carried through in neural network models in two ways. First, by introducing the BC in the loss function (soft manner)

and second, by using a predefined solution that automatically satisfies the boundary and initial conditions (hard manner). Applying the BCs by an ansatz in a hard manner is more favourable since it brings about more stability and fast convergence, mainly by decoupling the mapping data loss from the loss of residuals. This decoupling avoid the imbalanced distribution of loss gradients in backpropagation [24].

Few studies focused on the surrogate modelling of solutions to PDEs by imposing hard constraints on the deep networks. Nabian and Meidani [32] applied an ansatz, which included two terms, BC and DNN to solve high-dimensional stochastic conduction equation. Sun et al. [33] showed a hard enforcement application in surrogate modelling of Navier–Stokes equations by embedding the geometrical parameters of the regular domain and fluid properties into the network structure to enhance the model generalisation. Additionally, they compared the performance of hard and soft constraints. Their findings indicated the superiority of the hard enforcement approach in terms of solution accuracy. Furthermore, the PINN model with hard constraints has been applied to inverse problems; for instance, topology optimisation of fluids in a Stokes flow by Lu et al. [34]. Berg and Nyström [35] proposed an ansatz model, which employs NNs to estimate the BCs and distance functions in irregular geometries. They solved advection and diffusion problems in complicated geometries using this ansatz model.

Herein, our focus centres on the generalisation aspect of the surrogate model when it encounters domains of any pattern with no need for any measured data. We define a new solutions ansatz comprising a boundary function, a distance function and a DNN, which is a function of both aforementioned functions. In fact, the DNN is forced to learn the physics of the problem in accordance with the domain parameters. We evaluate the performance of this new ansatz model in terms of convergence and accuracy by solving the PDE of the heat conduction problem and comparing the results against those obtained via the previous model in the literature [35]. Furthermore, they are applied to heat conduction problems in different case studies, including 1D, 2D and 3D problems.

2. PINN methodology for solving PDEs

In this section, we provide a short overview of the PINN structure for solving the 2D steady-state conduction heat transfer with arbitrary Dirichlet boundary conditions. The PINN framework, in general, is based on the approximation of hidden solutions by a DNN, which is trained by data from the BC while respecting the governing equations given as:

$$f := \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) := \mathcal{N}[T(x, y)], (x, y) \in \Omega, \quad (1)$$

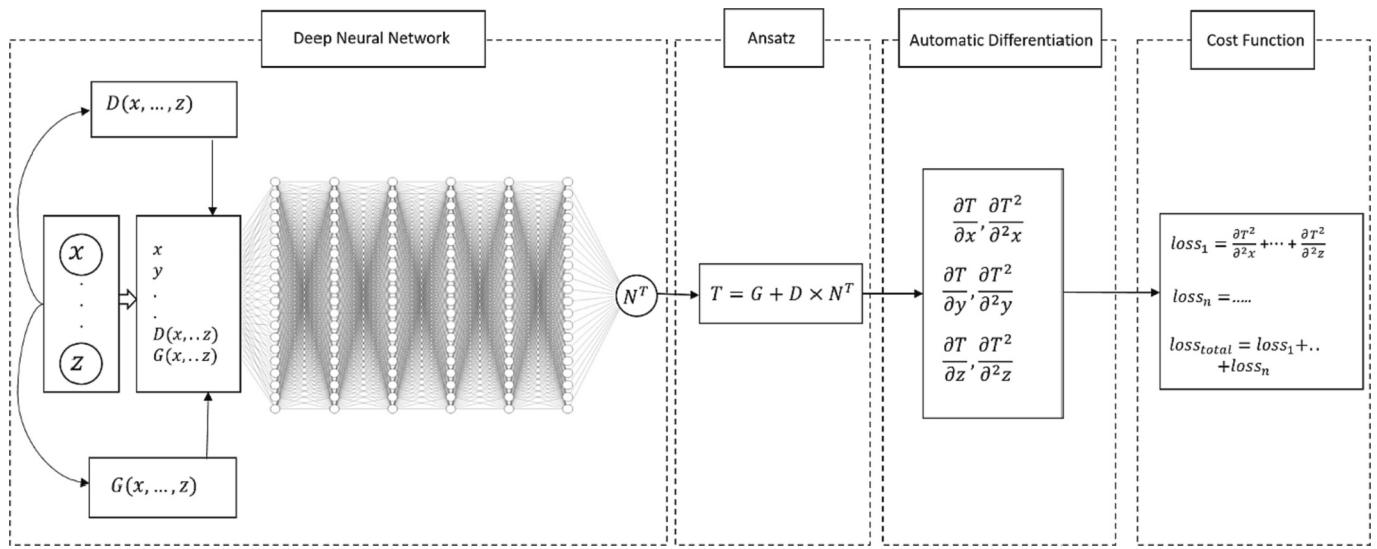


Fig. 1. Surrogate model workflow diagram with hard constraint.

$$T(x, y) = T_{BC}, (x, y) \in \Gamma,$$

where x and y are the spatial variables, $T(x, y)$ indicates the hidden solution and $\mathcal{N}[\cdot]$ is a nonlinear differential operator. $\Omega \subset \mathbb{R}^d$ is the domain over which we study the system. The domain is bounded by boundaries $\partial\Omega = \Gamma$ with corresponding known temperature (T_{BC}). The loss function of the PINN is the mean squared error of PINN predictions to the true values, defined as:

$$\text{loss}_{\text{total}} = \text{loss}_f + \text{loss}_{BC} \quad (2)$$

where,

$$\text{loss}_{BC} = \frac{1}{N} \sum_{i=1}^{N_b} |T_{BC} - N^T((x_i, y_i); \theta)|^2, (x_i, y_i) \in \Gamma, \quad (3a)$$

$$\text{loss}_f = \frac{1}{N_n} \sum_{i=1}^{N_n} |f((x_i, y_i); \theta)|^2, (x_i, y_i) \in \Omega \quad (3b)$$

The term loss_f denotes the residual error and loss_{BC} corresponds to the error in satisfying the BCs. This way of imposing BCs in NN models is known as the soft manner. $N^T((x_i, y_i); \theta)$ is the DNN that outputs temperature and, during the training, adjusts its parameters (θ) to minimise the total loss function ($\text{loss}_{\text{total}}$). The neural network architecture is described as two input variables for spatial quantities connected to the hidden layers and one output for temperature. It is noteworthy that terms regarding temperature derivatives to the inputs, which are required in calculating the loss function are computed during the training via AD.

In contrast to the soft manner, when the BCs are applied with hard constraints, an ansatz for the solution is defined that automatically satisfies the boundary and initial conditions. Below we present our ansatz model as Eq. (4):

$$T(x, y) = G(x, y) + D(x, y) * N^T([x, y, G, D]; \theta) \quad (4)$$

$G(x, y)$ is a smooth extension of boundary values that outputs temperatures. The general form of $G(x, y)$ does not matter as long as it projects the correct values at the boundaries. Moreover, it can either be described by a mathematical equation in the case of regular geometries or represented with a shallow or deep neural network for more complex geometries. In this study, the use of neural networks for producing the boundary condition function ($G(x, y)$) and distance function ($D(x, y)$) ensures the smoothness of both functions, which is mainly because of smooth activation functions, e.g. hyperbolic tangent, employed in the

structure of NNs.

To train the boundary condition NN ($G(x, y)$) we use the following loss function on the N_b nodes at the boundary:

$$MSE_G = \frac{1}{N_b} \sum_{i=1}^{N_b} |G(x_i, y_i) - T_{bc}(x_i, y_i)|^2, (x_i, y_i) \in \Gamma \quad (5)$$

$D(x, y)$ represents a smooth distance function of the collocation nodes to the boundaries. Its values on the boundaries are zero, ensuring that $T(x, y)$ obtains the correct values of the boundary function (Eq. (4)). To build the $D(x, y)$ function, first the minimum distance of each collocation node to the boundaries is computed via Euclidean distance, resulting in a non-smooth distance function $d(x, y)$. Then we fit a NN and train it to create a smooth distance function for all collocation nodes. The loss function of this neural network is calculated using the following equation.

$$d(x, y) = \min((x, y) - (x_{bc}, y_{bc})), (x, y) \in \Omega, (x_{bc}, y_{bc}) \in \Gamma,$$

$$MSE_D = \frac{1}{N_n} \sum_{i=1}^{N_n} |D(x_i, y_i) - d(x_i, y_i)|^2, (x_i, y_i) \in \Omega \quad (6)$$

Once both networks are trained, their parameters are set as untrainable (i.e. parameters are not updated further during the training of the $N^T([x, y, G, D]; \theta)$, and along with N^T (Eq. (4)) are embedded in the calculations of the loss functions. In our proposition, the input of $N^T([x, y, G, D]; \theta)$ includes terms regarding the BCs and distance function in its configuration beside input variables. In this way, the DNN is trained while respecting the domain features. Finally, the PINN model loss function shrinks as follows:

$$\text{loss}_{\text{total}} = MSE_f = \frac{1}{N_n} \sum_{i=1}^{N_n} |f(x_i, y_i)|^2, (x_i, y_i) \in \Omega \quad (7)$$

Fig. 1 demonstrates the paradigm of the PINN model integrated with the suggested ansatz model for solving PDEs. The spatiotemporal characteristics of the domain and two pre-trained functions (subnetworks), boundary values and distance functions, are the inputs of the DNN, forming an ansatz that is constrained by physical laws during the network training. In broad terms, the method we propose can be summarised in Algorithm 1.

Algorithm 1. Procedure The PINN algorithm for solving heat conduction equations with hard constraint.

(continued on next page)

(continued)

	Build and train two neural networks for boundary conditions $G(x,y)$ and distance function $D(x,y)$ in SciAnn as:
Step 1:	$D = \text{sn. Functional}'(D')$, variables = $[x,y]$, hidden_layers = $2 * [5]$, activation = ' $\tan h'$, output_activation = 'linear'
	$G = \text{sn. Functional}'(G')$, variables = $[x,y]$, hidden_layers = $2 * [10]$, activation = ' $\tan h'$, output_activation = 'linear'
Step 2:	Setting their parameters as untrainable $D.set_trainable(False)$ $G.set_trainable(False)$
Step 3:	Build the deep neural network and new ansatz model $N = \text{sn. Functional}'(N')$, variables = $[x,y,G,D]$, hidden_layers = $6 * [20]$, activation = 'tanh', output_activation = 'linear'
Step 4:	$T = G + D \times N([x,y,G,D], \theta)$
Step 5:	Specify the loss function using AD as: $\text{loss}(\theta) = \frac{\partial T^2}{\partial^2 x} + \frac{\partial T^2}{\partial^2 y}$ Train the neural network to find the best parameters θ^* by minimising the loss function $\text{loss}(\theta)$

3. Model implementation

Different application programming interfaces (APIs) have been introduced for building PINN models. Here we use SciAnn [36] a Python library that uses Keras and TensorFlow to build and train neural networks. All neural networks solving the PDEs for 1D and 2D case studies are fully connected feed-forward neural networks with six layers and 20 neurons in each layer (6 \times 20). This structure was selected in a trial-and-error manner. This was done by investigating the effect of different NN capacities (hidden layers and neurons) on the accuracy of predictions after a certain number of epochs. This structure was also used for the 3D problem, but its efficiency was not checked here. For training the NN of $D(x,y)$ and $G(x,y)$, the low-capacity neural network was identified to meet the loss convergence of 10^{-5} . The inputs of $G(x,y)$ and $D(x,y)$ are spatial parameters, and the inputs of the main DNN are spatial parameters along with outputs of $G(x,y)$ and $D(x,y)$. For the case of unsteady problems, the term regarding time can be added to the input variables. The hyperbolic tangent was assigned to the activation function of all layers except for the output layer, which is a linear function. There are different optimisation algorithms; we used Adam [37] optimiser, which is a modified stochastic gradient descent optimiser. Different parameters have an influence on the performance of the optimiser, such as learning rate and batch-size. In this study the loss function is minimised with an initial adaptive learning rate of 0.001, which decreases by half after 100 epochs in the case of no change in the loss values. The batch-size is 64, and the network convergence is controlled via the loss mean squared error.

4. Results

Here, the performance of the proposed ansatz is compared with the previous model proposed by Berg and Nyström [35] in 1D, 2D and 3D conduction-based heat transfer problems. To make our comparisons only dependent on the definition of the ansatz model, all network parameters involved in training the PINNs, e.g., the number of hidden layers, activation function, optimisation algorithms and epochs (number of iterations), are the same for both models.

4.1. One-dimensional case study

The first case study solves a 1D heat conduction problem in an infinite hollow cylinder with inner and outer radii r_1 and r_2 to investigate the function approximation capability of the proposed methodology. At steady state, the thermal diffusion in the cylindrical coordinates with azimuthal symmetry is described as follows:

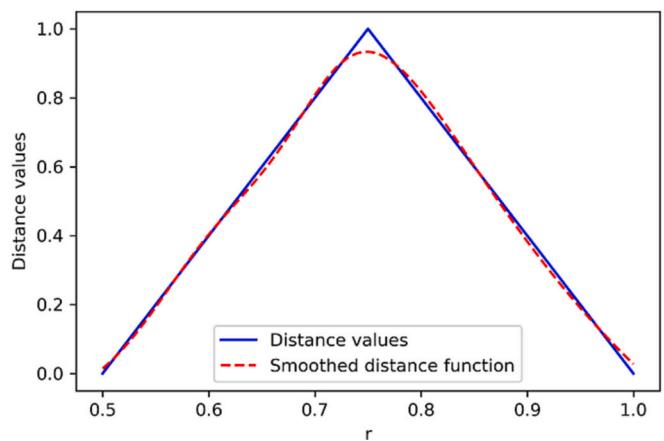


Fig. 2. Smoothed distance function for a 1D cylindrical problem.

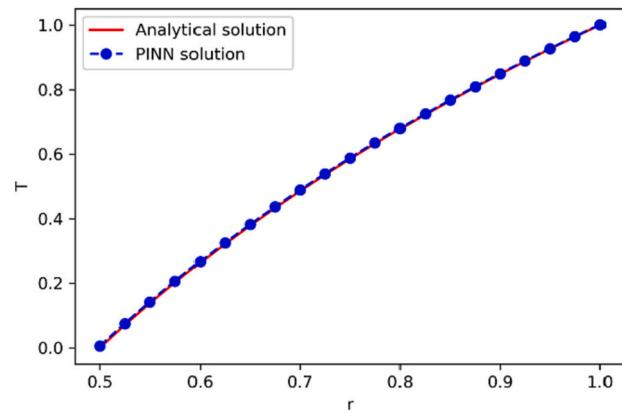


Fig. 3. PINN prediction and true values.

$$\begin{aligned} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) &= 0, 0.5 < r < 1 \\ T(0.5) &= 0 \\ T(1) &= 1 \end{aligned} \quad (8)$$

For the given BC, the analytical solution is determined as follows:

$$T(r) = C_1 \ln(r) + C_2,$$

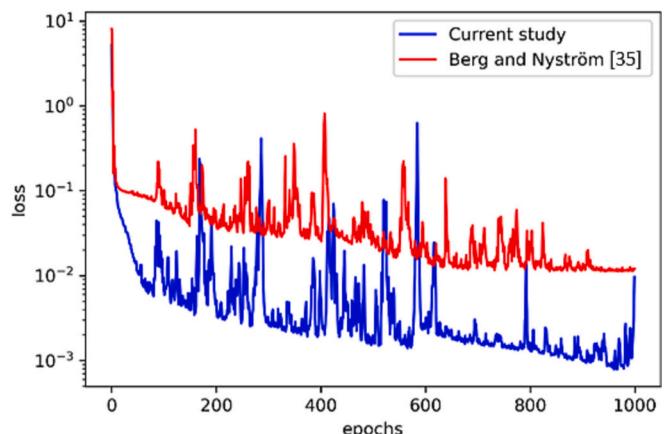


Fig. 4. Loss convergence of two models.

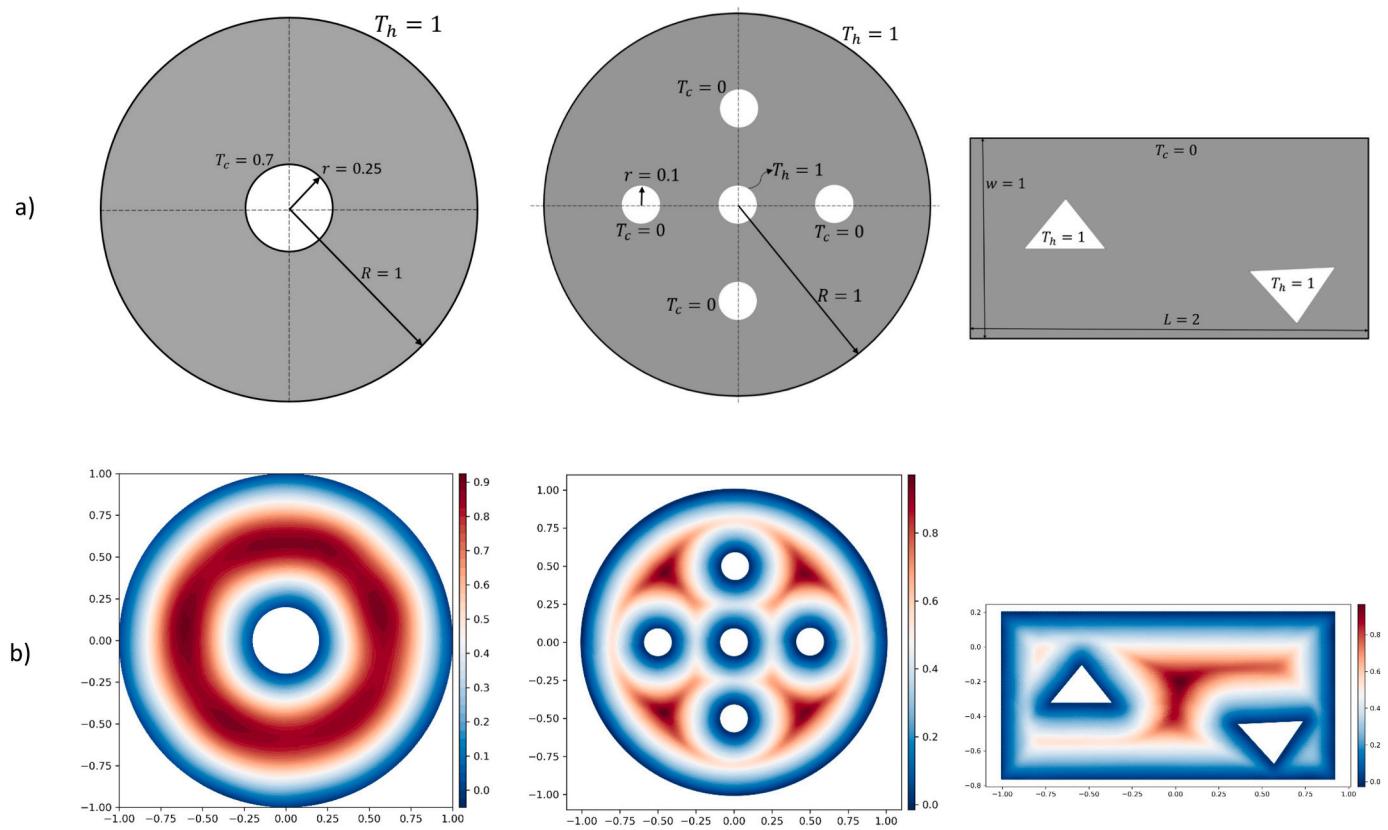


Fig. 5. The geometry of the problem and corresponding distance values. a) Three geometries of 2D problems with constant temperature boundary conditions. T_c is the lower temperature and T_h the higher temperature. b) The smoothed distance values calculated by the neural network.

$$C_1 = \frac{T(1) - T(2)}{\ln\left(\frac{r_1}{r_2}\right)}$$

$$C_2 = T(1) - C_1 \ln(r_1) \quad (9)$$

where T indicates the temperature and C_1 and C_2 are the constants of the analytical solution calculated according to the boundary conditions. This analytical solution serves as the basis for validating neural network estimations. In the first step, two networks of $D(r)$ and $G(r)$ are trained. Fig. 2 plots the distance values and the network smoothing on the distance values. The solution is presented in Fig. 3, which uses the PINN model at 1000 epochs and an analytical approach. Evidently, a perfect match exists between the two results. To assess the performance of the new ansatz model, the loss convergence rates of both models in Fig. 4 were compared. Obviously, the PINN model integrated with the new ansatz model converges faster than its counterpart to loss values, which are lower by one order of magnitude.

Here, 2D steady-state heat conduction problems in three geometries with constant temperatures are investigated. The ground truth-values are the simulation results of the finite volume method (FVM) implemented in the commercial software Ansys Fluent Release 2020 R2. The heat transfer equation is solved on meshes generated via the Ansys Meshing module using the quadrilateral dominant method for 2D cases, and for the 3D geometry, all grids are composed of hexahedral elements. Furthermore, the accuracy of the numerical simulation was tested by comparing them with the analytical solution for the geometry of concentric circles. An MSE between the CFD result and the analytical solution equal to 10^{-7} was obtained, which verifies the efficacy of CFD solvers; hence, it was used for the rest of the case studies in this study.

Fig. 5 shows the domains and BCs of each case study at the top and the corresponding outputs of the smoothed distance function below. In all case studies, the Dirichlet BC is imposed on the walls, where T_h and

T_c indicate the hot and cold temperatures, respectively. The geometries' dimensions are normalised by dividing the lengths by the maximum length. In addition, the distance values are normalised between zero and one.

4.1.1. Nodal distribution study

The number of nodes and node distribution are other important factors in training neural network models and influence the accuracy of predictions. Therefore, it is necessary for the NN models solution to be independent of the number of nodes and the distribution. Since no labelled data from collocation points (interior domain) is available and there is no preference for sample (node) selection, there is uniform distribution for all case studies. Here the model performance is examined using three different sample densities of 894, 3309 and 19,836 for the case study of a rectangular domain containing two triangular objects. The distribution of the nodes is depicted in Fig. 6, where the temperature of triangular objects and external wall are considered one and zero, respectively. As a first step, neural network models for mapping data (boundary conditions and distance values) are built, which will be deployed in the structure of the predefined solution afterwards. The results showed that training models with fewer nodes for data mapping tasks leads to lower accuracy, whereas the high accuracy of order 10^{-5} was obtained with higher densities for the same number of iterations. In the next step, the governing equations are solved with the PINN model for three samples numbers, by minimising the loss residual of the conduction equation. It was observed that training the PINN model with 894 nodes converged faster than training with 19,836 nodes and even achieved smaller loss values but it could not guarantee the accuracy of the results. This justifies the necessity of model validation in addition to monitoring loss convergence. To evaluate results accuracy locally, the temperature distributions of different sample densities, along with the CFD results at a specific height, are plotted in Fig. 7.

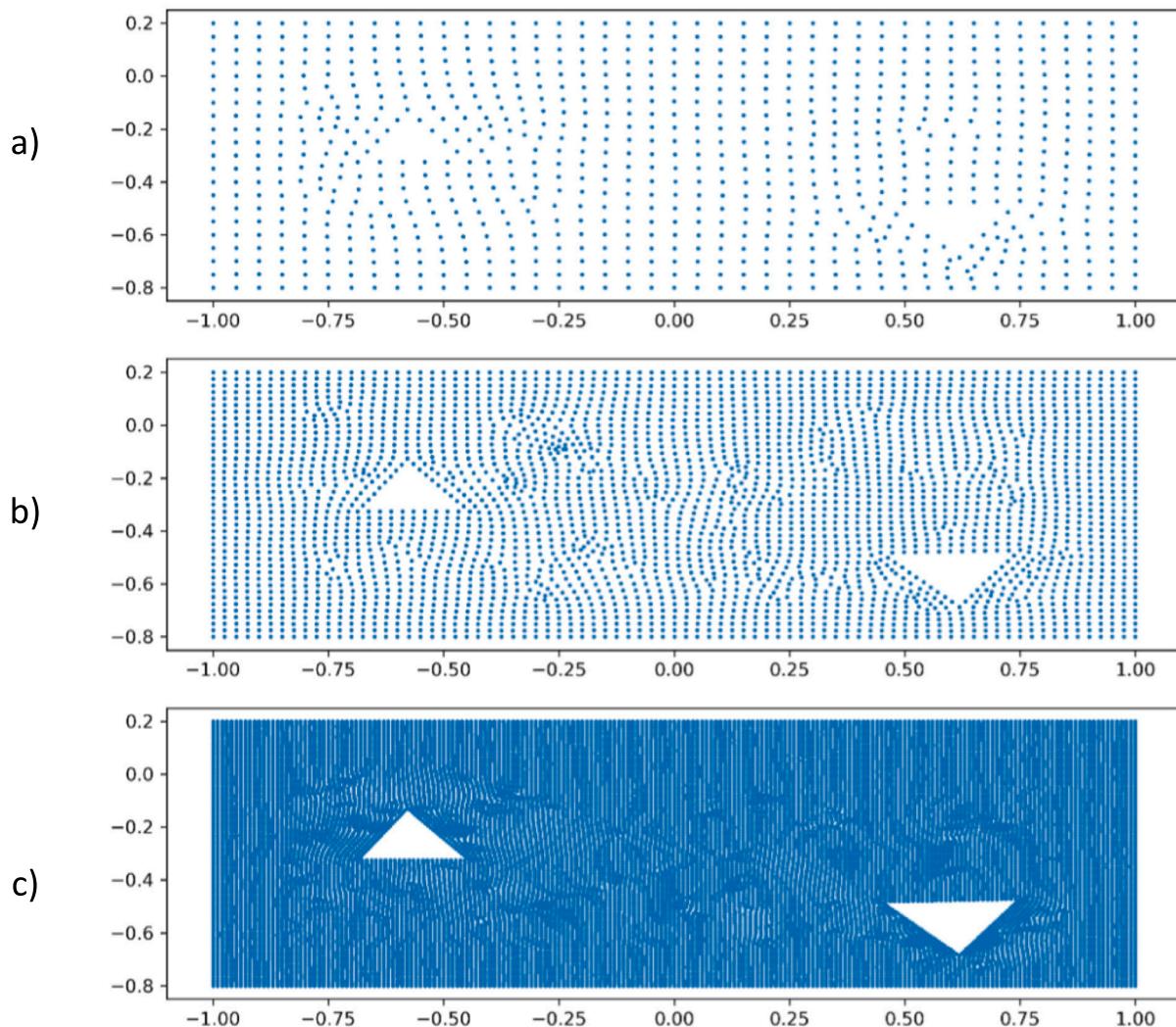


Fig. 6. Node distribution of three densities. a) 894 nodes. b) 3309 nodes. c) 19,836 nodes.

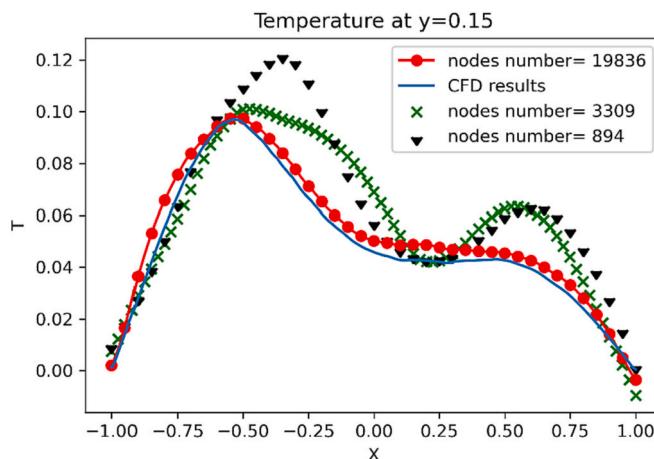


Fig. 7. The effect of the number of nodes on the temperature estimation.

Furthermore, the MSE that indicates the model performance for the entire domain is calculated for each density and presented below in **Table 1**. Looking at both **Fig. 7** and **Table 1**, we can conclude that

Table 1
The mean squared error for different N nodes.

Number of nodes	Nodes on the Boundaries	MSE
894	151	0.00713
3309	302	0.00220
19,836	753	0.00055

increasing the number of nodes improves the accuracy of the results and model performance. However, it also increase the computational costs as well; thus, a trade-off between accuracy and computational time needs to be considered.

4.1.2. 2D results

Fig. 8 summarises the temperature distribution predicted by the PINN integrated with the new ansatz model for three case studies. It is followed by values acquired via the CFD study, which serves as the truth values, and the corresponding relative errors. The 2D examples are relatively more complicated than the 1D problem, which are computationally more cost-intensive. Thus, the PINN model was trained for the first 2D case study with 1000 epochs and the second and third geometries, including sharp angles and more details with 10,000 and 30,000 epochs.

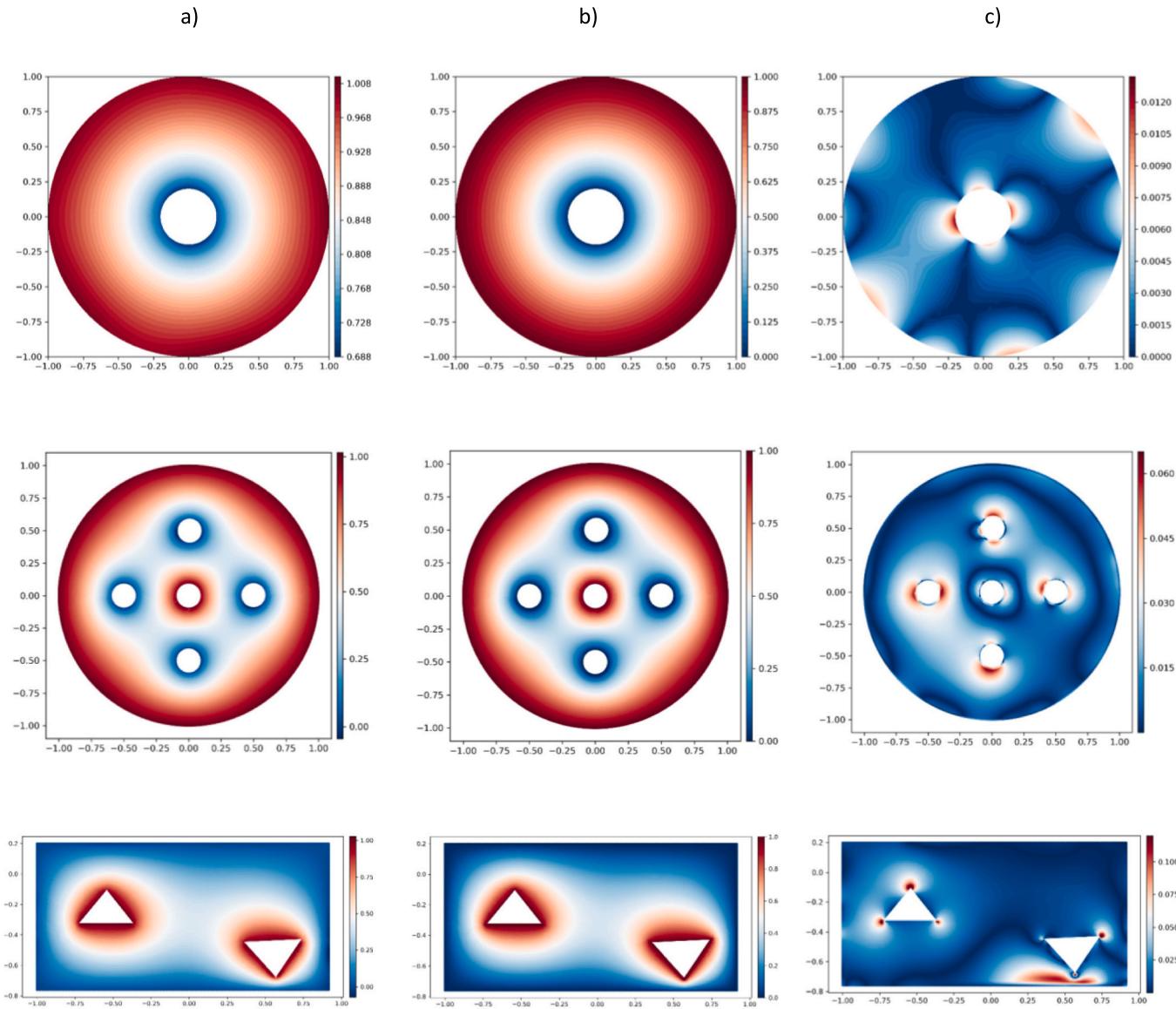


Fig. 8. Temperature distribution in three case studies. a) The first column depicts the result of the PINN model. b) The second column shows the result of the CFD simulation. c) The third column is the relative error between the results of the PINN model and CFD results.

Furthermore, the performance of the two ansatz models was quantitatively compared in terms of temperature distribution estimation and loss convergence. Fig. 9. shows the results of the temperature estimation at 1000 epochs, and $y = 0.25$ for a concentric geometry where the cold temperature of the inner circle is 0.7 and the outer circle temperature is 1. Note that both models were trained with the same neural network structures and parameters. It can be seen that both models outcomes are consistent with the CFD results, while the loss convergence of the PINN model based on the new model tends to converge faster to loss values which are lower by one order of magnitude with respect to the literature-based model [35].

The same behavioural characteristics were observed in the two subsequent case studies, aiming to solve the conduction-based heat transfer in the more complex geometry to explore the role of geometrical features in training models and possible bottlenecks. Fig. 10 shows the loss convergence of the two models for the geometry of multiple cylindrical objects in a cylinder (Fig. 10.a) and a rectangular domain containing two triangular objects (Fig. 10.b), where one of triangular objects was placed near the wall intentionally to examine the narrow

gap between two boundaries. According to our observations, the geometry with a sharp angle delayed the convergence while the same physics was solved. As it is shown in Fig. 8-c, maximum error happens on the sharp edges and where the gap between the objects and wall is narrow. The comparison of the mean squared error of both models showed that the new ansatz model reduced the prediction error from 0.000823 to 0.000652 for the circular domain and from 0.001748 to 0.000534 for the rectangular domain.

The model accuracy is analysed by monitoring the results of the two models compared to the CFD study at 10000 and 30,000 epochs for the case of a rectangular domain containing two triangular objects. Fig. 11-a shows that the temperature estimation at $y = -0.01$ using the previous model (Berg and Nyström [35]) deviates largely from the CFD results at 15000 epochs with respect to the solution of the present study. Fig. 11-b demonstrates the solutions at 30000 epochs, where the result of the new model matches the ground-truth values, showing good accuracy with lower computational effort. Our results show that the new model improves the convergence speed, which is a bottleneck for current application of PINN models. The previous model converges to a similar

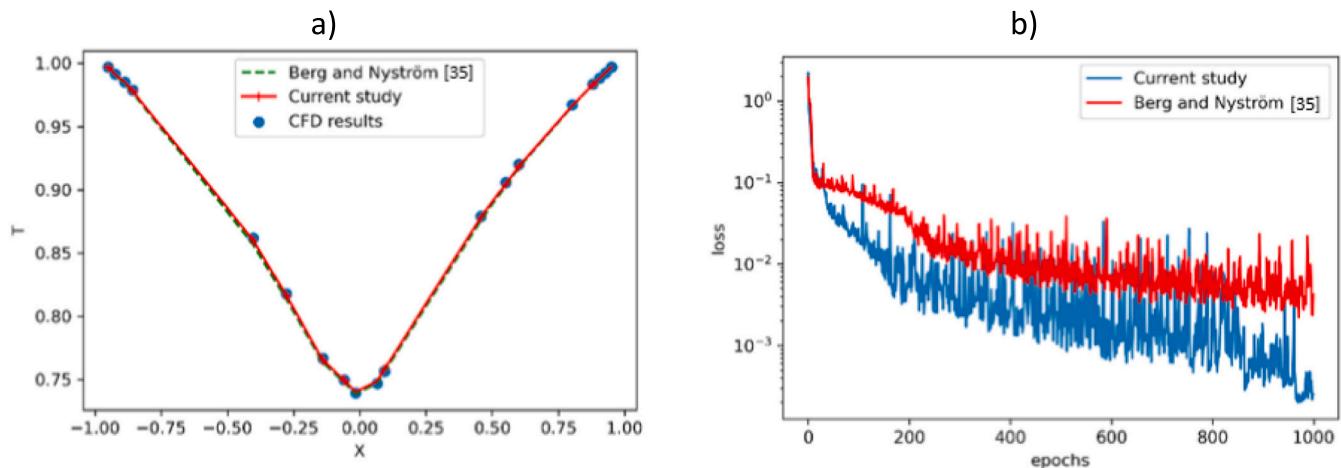


Fig. 9. The evaluation of the model performance for concentric geometry. a) Temperature estimation at $y = 0.25$. b) The loss comparison between the two models.

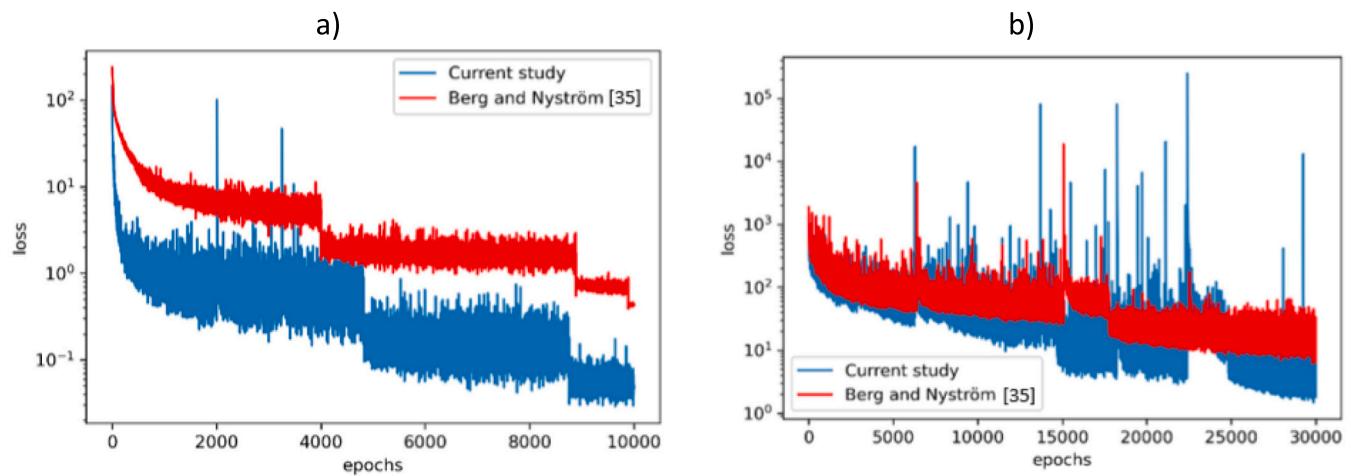


Fig. 10. Comparison of loss values of different case studies. a) Multiple cylindrical objects in a cylinder b) Rectangular domain containing two triangular objects.

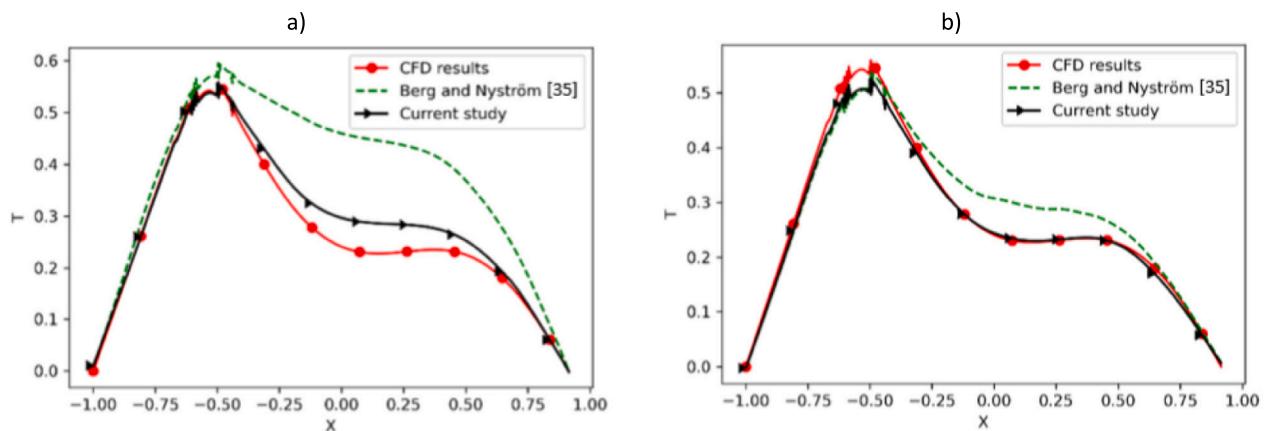


Fig. 11. Comparison of temperature estimation of neural network models and CFD study at different epochs. The red line with circle symbol indicates the CFD results, the green dashed line and black line with arrow symbols represent the model in the literature (Berg and Nyström) and the current study. a) epochs = 15,000 b) epochs = 30,000 (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

accuracy after 110,000 epochs.

4.2. Two-dimensional case study with hybrid constraint

The third case study explores solving heat conduction problems with Dirichlet and Neumann BCs. The critical aspect of implementing the

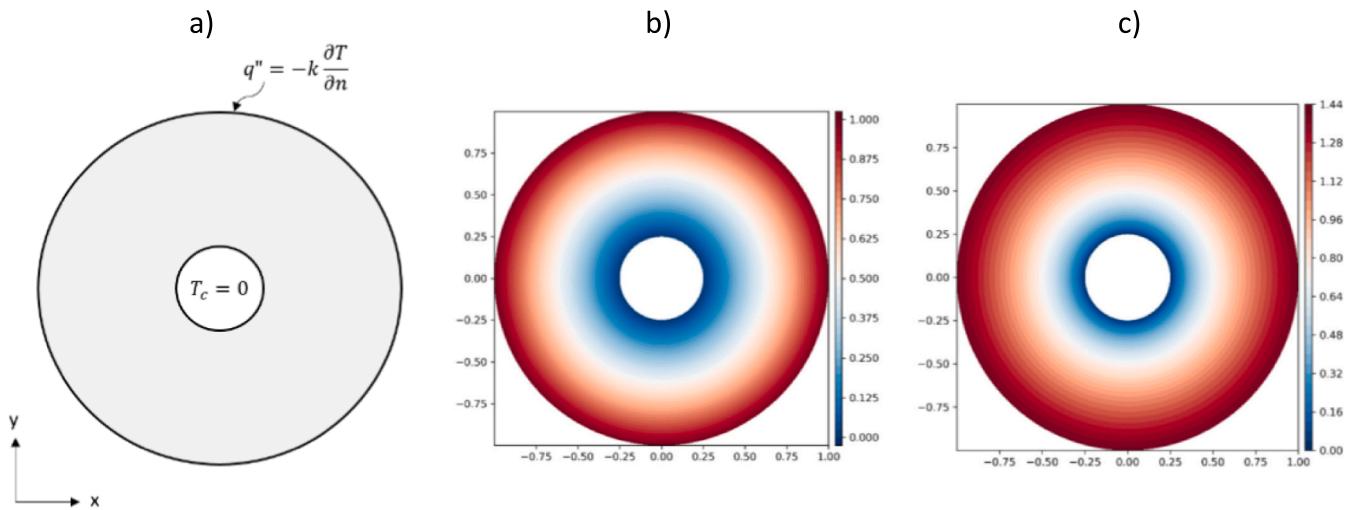


Fig. 12. Heat conduction problem with different BCs. a) Geometry and BC of problem. b) Distribution of distance values. c) PINN prediction.

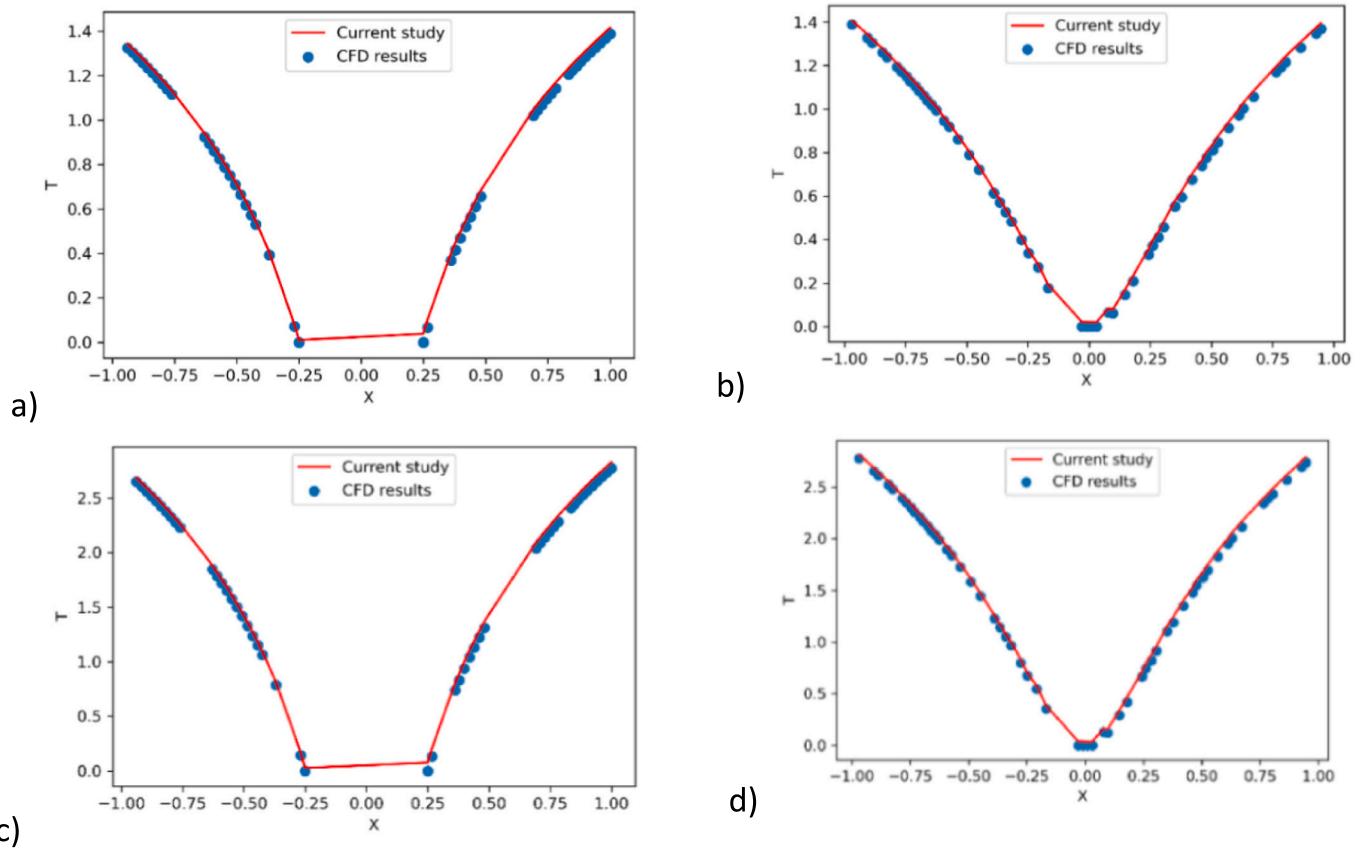


Fig. 13. Temperature distributions under different heat fluxes. a) $y = 0$ and $q'' = 1 \text{ (}\frac{w}{m^2}\text{)}$. b) $y = 0.25$ and $q'' = 1 \text{ (}\frac{w}{m^2}\text{)}$. c) $y = 0$ and $q'' = 2 \text{ (}\frac{w}{m^2}\text{)}$. d) $y = 0.25$ and $q'' = 1 \text{ (}\frac{w}{m^2}\text{)}$.

Neumann BC in a purely complicated manner concerns the association of the boundary values with the values from the domain interior, which cannot be determined by training the BC neural network beforehand. Consequently, it is not feasible to enforce a Neumann BC via a hard constraint. Hence, we apply a hybrid constraint in problems with different BCs so that the Dirichlet and Neumann BCs are imposed with hard and soft enforcements, respectively. The predefined solution is the same as before (Eq. (4)), and the $G(x, y)$ function estimates the values of N_D Dirichlet boundary nodes, whereas the loss of PINN includes terms

regarding the residual loss of N_n collocation nodes and N_{Ne} Neumann boundary nodes.

$$\text{MSE}_G = \frac{1}{N_D} \sum_{i=1}^{N_D} |G(x_i, y_i) - T_{bc}|^2, (x_i, y_i) \in \Gamma_{Dirichlet}, \quad (10)$$

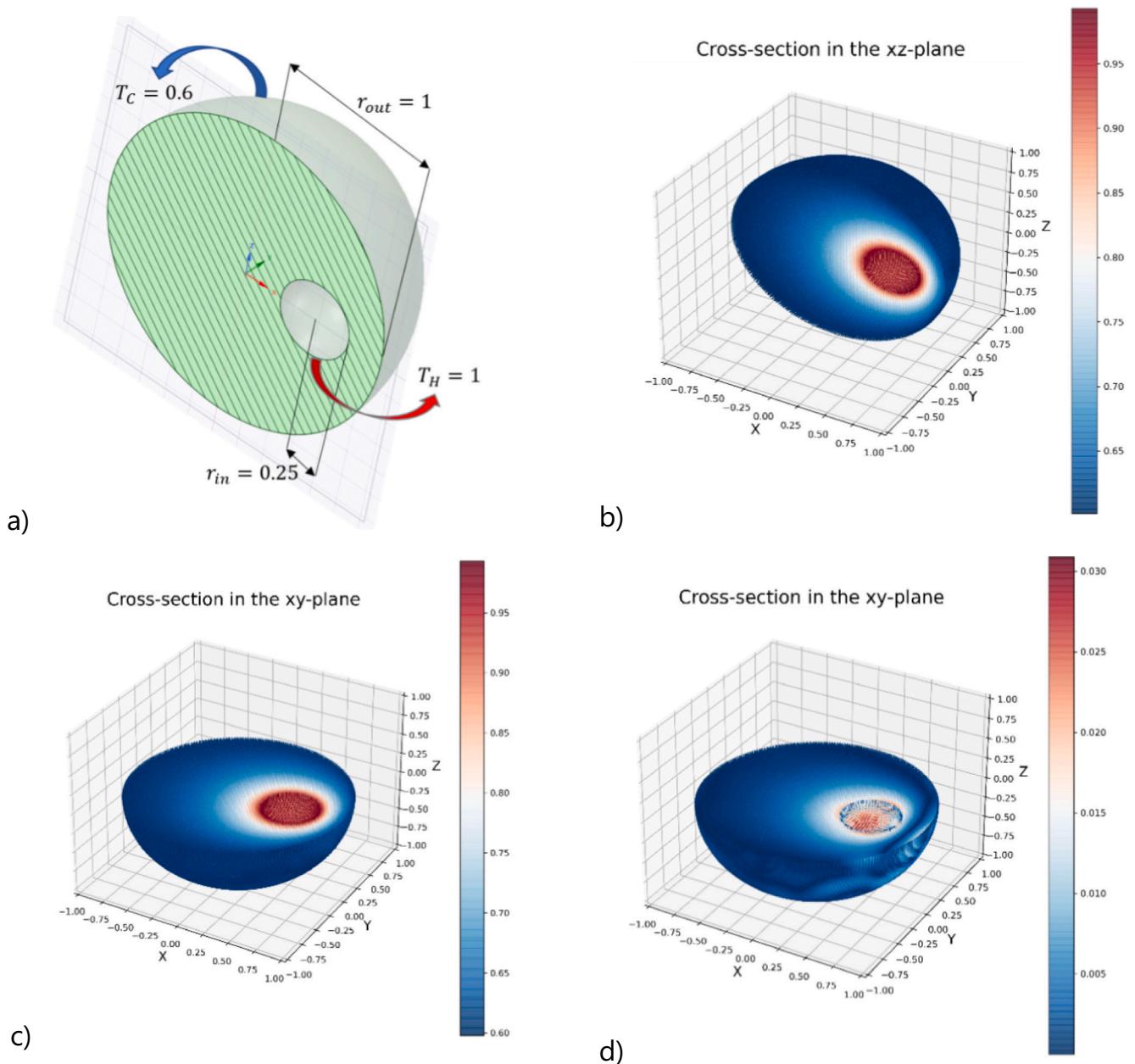


Fig. 14. a) the geometry of problem and boundary conditions in cross-section in the xz plane. b,c) PINN predictions on different cross-sections d) relative errors on a cross-section in the xy plane.

$$\begin{aligned} loss_{total} &= MSE_f + \lambda MSE_{Neumann\ BC} = \frac{1}{N_n} \sum_{i=1}^{N_n} |f(x_i, y_i)|^2, (x_i, y_i) \\ &\in \Omega + \lambda \sum_{i=1}^{N_{Ne}} |\mathbf{n} \bullet \nabla T(x_i, y_i) - B_{Neumann}|^2, (x_i, y_i) \in \Gamma_{Neumann} \end{aligned} \quad (11)$$

Here, λ is the relative weighting coefficient used for tuning unbalanced gradients by normalising the gradient of individual terms in the loss function to reduce the stiffness of the gradient flow dynamics [24]. And $B_{Neumann}$ is the Neumann boundary condition operator. Fig. 12 depicts the geometry of the problem, BCs and distance function for a heat conduction problem, where the inner circle is maintained at a low temperature, and the outer circle is under constant heat flux $1 (\frac{w}{m^2})$. Here, the distance function approximates the distances of collocation nodes from boundaries corresponding to only the Dirichlet BC. Thermal

conductivity is assumed to be $1 (\frac{w}{m K})$, and the low temperature (T_C) is considered 0.

Fig. 13 plots the temperature distributions along the x -axis at $y = 0$ and 0.25 with heat flux equal to 1 and $2 (\frac{w}{m^2})$. The results show the capability of the new ansatz model in handling different types of BCs as well. The accuracy of the results is examined against the numerical results of the CFD study, which shows good agreement between the results.

4.3. Three-dimensional case study

Developing three-dimensional (3D) PINN models tailored to real-world applications is of great interest to researchers. That is the motivation for the last case study that discusses the capability of the PINN model to deal with 3D problems and presents a comparison between the performance of two PINN models. Building a 3D model in the PINN

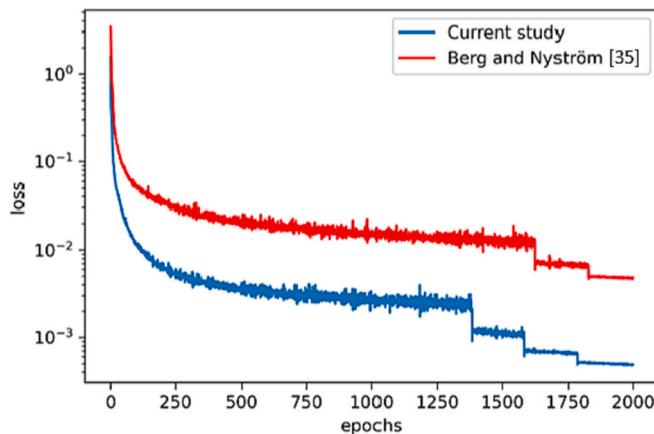


Fig. 15. Loss convergence of the 3D problem at epochs = 2000.

framework is carried out by adding a third dimension to spatial variables and easily reformulating the governing equations in the loss function in the form of 3D equations. Solving a 3D steady-state conductive heat transfer between two spherical shells is the target of this section. The 3D PINN structure is similar to the 2D models, while an input variable (z) in the direction of the third dimension is added to the inputs variables along with adjustments in the governing equation and corresponding derivative terms ($\frac{\partial^2 T}{\partial z^2}$). A cross-sectional view of the geometry, boundary conditions and the solution of the PINN model along with the relative error are illustrated in Fig. 14. The total number of nodes is 667,962, consisting of boundary nodes and interior nodes. The relative error between the PINN predictions and CFD values shown in Fig. 14-d, indicates a good consistency of outcomes with truth-values. Moreover, the new ansatz model performance, in comparison with the literature model [35] showed significant improvement in convergence speed (Fig. 15), by reducing the relative errors from 0.0041 to 0.0038 after the same number of iterations (2000).

5. Discussion and concluding remarks

Applying machine learning techniques to forward problems with no labelled data is a new field of study. However, its efficiency at this point is not comparable with conventional CFD methods that have matured well over the past decades. On the contrary, once the dynamic of the problem is bound up with parameters such as material and geometrical characteristics, conventional numerical methods appear to be inefficient, owing to the repetition of the entire simulation process. This issue is addressed via surrogate modelling that considers the effect of the desired dynamics in the training process. Moreover, the significance of surrogate modelling includes further applications in heat conduction systems, such as uncertainty quantification, optimisation design and inverse analysis. This way, once a model has been trained, it will make estimations for a wide range of given input values with no computational costs. To this end, a robust surrogate model that solves the heat equation is a prerequisite, which can be tailored afterwards for the intended targets. This was the main objective of this contribution, which introduced a new predefined solution for solving PDEs by efficiently using PINNs. According to our findings, The geometric shape was shown to be an influential factor in training cost, where geometries with rounded boundaries were more easily captured by the neural network, whereas the geometry with sharp corners tended to converge less.

In conclusion, the new model performance has been tested thoroughly on heat conduction problems under various conditions and geometries for 1D, 2D and 3D case studies. Furthermore, the node distribution study revealed that training the model with low densities might converge faster and to lower loss values; however, it does not represent the accuracy of predictions. Hence, the comparison against

truth-values was considered in all case studies to have a reliable evaluation on the model performance. It was observed that the proposed ansatz model hastened the convergence to a loss value, which is lower by one order of magnitude compared with the literature-based model with the same number of iterations along with better prediction accuracy. We have also formulated the hybrid constraint to address problems with Neumann and Dirichlet BCs using the new ansatz model for different heat fluxes. Further improvements could be made to reduce the computational costs and boost the model accuracy, such as applying adaptive activation functions, using second-order optimisation algorithms and the adaptive learning rate method, which have been excluded here. Our ongoing research will focus on solving problems with coupled and multi-physics governing equations using the new ansatz model introduced in this study.

CRediT authorship contribution statement

Seyedalborz Manavi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Thomas Becker:** Resources, Writing – review & editing, Funding acquisition. **Ehsan Fattahi:** Conceptualization, Methodology, Validation, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research project is part of the strategic alliance “WiPro - Wissensbasierter Prozessintelligenz” (Knowledge-Based Process Intelligence) and was supported by the German Ministry of Education and Research as part of the cross-alliance network “UfIB - Umsetzungsfördernde Initiative Bioökonomie” (Implementation-Promoting Initiative Bioeconomy).

References

- [1] F. Wang, M. Safdar, B. Jamil, M.I. Khan, S. Taj, M.Y. Malik, A.S. Alqahtani, A. M. Galal, One-dimensional optimal system of lie sub-algebra and analytic solutions for a liquid film fluid flow, *Chin. J. Phys.* 78 (2022) 220–233.
- [2] T. Lida, M. Minoura, Analytical Solution of Impulse Response Function of Finite-depth Water Waves 249, 2022, p. 110862.
- [3] V. Riego, M. Castejón-Limas, L. Sánchez-González, L. Fernández-Robles, H. Perez, J. Diez-Gonzalez, Á.-M. Guerrero-Higueras, Strong classification system for wear identification on milling processes using computer vision and ensemble learning, *Neurocomputing* 456 (2021) 678–684.
- [4] I. Lauriola, A. Lavelli, F. Aiolfi, An introduction to deep learning in natural language processing: models, techniques, and tools, *Neurocomputing* 470 (2021) 443–456.
- [5] S.M. Yusuf, F. Zhang, M. Zeng, M. Li, DeepPPF: a deep learning framework for predicting protein family, *Neurocomputing* 428 (2021) 19–29.
- [6] M.A. Álvarez, D. Luengo, N.D. Lawrence, Linear latent force models using Gaussian processes, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013) 2693–2705.
- [7] S. Thaler, L. Paehler, N.A. Adams, Sparse identification of truncation errors, *J. Comput. Phys.* 397 (2019), 108851.
- [8] S.L. Brunton, Applying machine learning to study fluid mechanics, *Acta Mech. Sinica* 37 (2022) 1718–1726.
- [9] G.V. Iungo, F. Viola, U. Ciri, M.A. Rotea, S. Leonardi, Data-driven RANS for simulations of large wind farms, *J. Phys. Conf. Ser.* 625 (2015) 12025.
- [10] J.-X. Wang, H. Xiao, Data-driven CFD modeling of turbulent flows through complex structures, *Int. J. Heat Fluid Flow* 62 (2016) 138–149.
- [11] B. Kim, V.C. Azevedo, N. Thuerey, T. Kim, M. Gross, B. Solenthaler, Deep fluids: a generative network for parameterized fluid simulations, *Comp. Graph. Forum* 38 (2019) 59–70.

- [12] L. Lu, P. Jin, G.E. Karniadakis, DeepONet: Learning Nonlinear Operators for Identifying Differential Equations Based on the Universal Approximation Theorem of Operators, 8 October 2019.
- [13] S. Goswami, K. Kontolati, M.D. Shields, G.E. Karniadakis, Deep Transfer Learning for Partial Differential Equations under Conditional Shift with DeepONet, 21 April 2022.
- [14] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier Neural Operator for Parametric Partial Differential Equations, 18 October 2020.
- [15] M. Edalatifar, M.B. Tavakoli, M. Ghalambaz, F. Setoudeh, Using deep learning to learn physics of conduction heat transfer, *J. Therm. Anal. Calorim.* 146 (2021) 1435–1452.
- [16] B. Kwon, F. Ejaz, L.K. Hwang, Machine learning for heat transfer correlations, *Int. Commun. Heat Mass Transf.* 116 (2020), 104694.
- [17] Q. Lin, Z. Liu, J. Hong, Method for directly and instantaneously predicting conductive heat transfer topologies by using supervised deep learning, *Int. Commun. Heat Mass Transf.* 109 (2019), 104368.
- [18] M. Wang, H. Wang, Y. Yin, S. Rahardja, Z. Qu, Temperature field prediction for various porous media considering variable boundary conditions using deep learning method, *Int. Commun. Heat Mass Transf.* 132 (2022), 105916.
- [19] A. Karpatne, G. Atluri, J. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: a new paradigm for scientific discovery from data, *IEEE Trans. Knowl. Data Eng.* 29 (2017) 2318–2331.
- [20] Jian Sun, Zhan Niu, Kristopher A. Innanen, Junxiao Li, Daniel O. Trad, A Theory-guided Deep Learning Formulation and Optimization of Seismic Waveform Inversion, 2022.
- [21] T. Zhang, S. Sun, Thermodynamics-informed neural network (TINN) for phase equilibrium calculations considering capillary pressure, *Energies* 14 (2021) 7724.
- [22] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [23] A.D. Jagtap, K. Kawaguchi, G. Em Karniadakis, Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks, in: *Proceedings Mathematical, Physical, and Engineering Sciences* 476, 2020, p. 20200334.
- [24] S. Wang, Y. Teng, P. Perdikaris, Understanding and Mitigating Gradient Pathologies in Physics-Informed Neural Networks, 13 January 2020.
- [25] K. Shukla, P.C. Di Leoni, J. Blackshire, D. Sparkman, G.E. Karniadakis, Physics-Informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks, 4 August 2020.
- [26] H. Eivazi, M. Tahani, P. Schlatter, R. Vinuesa, Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations, *Phys. Fluids* 34 (2022) 75117.
- [27] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, *J. Heat Transf.* 143 (2021) 2329.
- [28] S. Amini Niaki, E. Haghightat, T. Campbell, A. Poursartip, R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, *Comput. Methods Appl. Mech. Eng.* 384 (2021), 113959.
- [29] Z. He, F. Ni, W. Wang, J. Zhang, A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials, *Mat. Today Commun.* 28 (2021), 102719.
- [30] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial Neural Networks for Solving Ordinary and Partial Differential Equations 9, 1997, pp. 987–1000.
- [31] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind, Automatic Differentiation in Machine Learning: A Survey 18, 2018, pp. 1–43.
- [32] M.A. Nabian, H. Meidani, A deep neural network surrogate for high-dimensional random partial differential equations, *Prob. Eng. Mech.* 57 (2019) 14–25.
- [33] L. Sun, H. Gao, S. Pan, J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, *Comput. Methods Appl. Mech. Eng.* 361 (2020), 112732.
- [34] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S.G. Johnson, Physics-Informed Neural Networks with Hard Constraints for Inverse Design, 9 February 2021.
- [35] J. Berg, K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, *Neurocomputing* 317 (2018) 28–41.
- [36] E. Haghightat, R. Juanes, SciANN: a Keras/Tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, *Comput. Methods Appl. Mech. Eng.* 373 (2021), 113552.
- [37] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 22 December 2014.