A physics-driven and machine learning-based digital twinning approach to transient thermal systems

Armando Di Meglio and Nicola Massarotti
Engineering Department, University of Naples Parthenope, Naples, Italy, and
Perumal Nithiarasu
Zienkiewicz Institute for Modelling, Data and AI, Swansea University, Swansea, UK

Received 16 October 2023
Revised 21 January 2024
7 February 2024
Accepted 8 February 2024

Abstract

Purpose – In this study, the authors propose a novel digital twinning approach specifically designed for controlling transient thermal systems. The purpose of this study is to harness the combined power of deep learning (DL) and physics-based methods (PBM) to create an active virtual replica of the physical system.

Design/methodology/approach – To achieve this goal, we introduce a deep neural network (DNN) as the digital twin and a Finite Element (FE) model as the physical system. This integrated approach is used to address the challenges of controlling an unsteady heat transfer problem with an integrated feedback loop.

Findings – The results of our study demonstrate the effectiveness of the proposed digital twinning approach in regulating the maximum temperature within the system under varying and unsteady heat flux conditions. The DNN, trained on stationary data, plays a crucial role in determining the heat transfer coefficients necessary to maintain temperatures below a defined threshold value, such as the material's melting point. The system is successfully controlled in 1D, 2D and 3D case studies. However, careful evaluations should be conducted if such a training approach, based on steady-state data, is applied to completely different transient heat transfer problems.

Originality/value – The present work represents one of the first examples of a comprehensive digital twinning approach to transient thermal systems, driven by data. One of the noteworthy features of this approach is its robustness. Adopting a training based on dimensionless data, the approach can seamlessly accommodate changes in thermal capacity and thermal conductivity without the need for retraining.

Keywords Digital twinning, Digital twin (DT), Transient heat transfer, FEM, Deep learning, Machine learning, Data exchange

Paper type Research paper

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 34 No. 6, 2024
pp. 2229-2256
Emerald Publishing Limited
0961-5539
DOI 10.1108/HFF-10-2023-0616

Nomenclature

Symbol	Description
AI =	Artificial intelligence
ML =	Machine learning
DL =	Deep learning
DT =	Digital twin
FEM =	Finite element method
(D)NN =	(Deep) neural network
SL =	Supervised learning
DRL =	Deep reinforcement learning
CAD =	Computer-aided design
CAE =	Computer-aided engineering
MSE =	Mean square error
RMSE =	Root mean square error
MAPE =	Mean absolute percentage error
melt	Melting
ReLU =	Rectified linear unit
Fo =	Fourier number
Bi =	Biot number
T =	Temperature
k =	Thermal conductivity
ρ =	Density
c =	Specific heat
t =	Time
q =	Heat flux
h_c =	Heat transfer coefficient
L =	Length
α =	Thermal diffusivity
y =	Coordinate or prediction

1. Introduction

In our rapidly evolving technological landscape, the field of artificial intelligence (AI) has gained paramount significance. AI is a discipline focused on imparting machines with pseudo-cognitive capacities to mimic human intelligence. Within this vast domain, machine learning (ML) is a sub-set concerned with the development of algorithms that allow systems to learn and adapt from data without explicit programming. Deep learning (DL), in turn, stands as a specialized branch of ML, characterized by the utilization of deep neural networks (DNNs) to enable high-level pattern recognition and complex feature extraction (Ongsulee, 2017). Alongside these advancements, the concept of the digital twin (DT) has emerged as a revolutionary framework that may amalgamate these technologies to offer a real-time digital copy of a physical entity or system. The DT concept embodies various definitions, each tailored to its application domain. However, a common thread among these definitions is that the DT is a digital mapping of the physical world aimed at facilitating analysis, prediction, and especially control of the physical world itself (Yao *et al.*, 2023).

From a theoretical perspective, a DT can exist independently of the utilization of AI and its sub-sets. A DT can be based on CAD (computer-aided design) models, CAE (computer-aided engineering) simulations and the like. However, a challenge arises due to the fact that traditional 2D/3D simulations are still distant from being capable of exchanging real-time data with the physical system. To address this issue, AI, especially ML and DL, becomes a

highly useful tool for constructing predictive reduced models of the real system (Xuereb Conti *et al.*, 2023; Grabe *et al.*, 2023), optimization purposes (Lin *et al.*, 2018; Guo *et al.*, 2023; Keramati *et al.*, 2022; Aldaghi *et al.*, 2023; Zeeshan *et al.*, 2023), solving inverse problems (Tamaddon-Jahromi *et al.*, 2020; Löhner *et al.*, 2020) and effectively mitigating the computational costs associated with conventional simulations (Zhu *et al.*, 2023; Park and Kim, 2023; Zhang *et al.*, 2023). It must be specified that the standalone CAE simulations do not represent a DT. Yao *et al.* (2023), in their review work, highlight the main difference between CAE simulations and DTs. Quoting from their work: "Simulation technologies partially reproduce the real world offline, primarily during the research and design stages. They usually do not perform analyses or optimization functions. However, DT reflects the state changes of physical objects in real-time and can be used to analyse and predict the decision optimization function of physical entities. Simulation technology relies on models and data to map the properties and parameters of the physical world. DT must sense, diagnose, and predict the state of physical entities in real time to optimize them".

DNNs, a cornerstone of DL, find relevance in the creation of DTs by enabling the synthesis of intricate models capable of replicating complex behaviours. Supervised learning (SL), a sub-set of ML, contributes to training these networks, establishing a connection between AI and DTs. Furthermore, the realm of deep reinforcement learning (DRL) extends this interconnection, where AI agents learn to make decisions through interactions with their environment, mirroring the way control strategies can be developed and optimized within the DT framework.

The history of DT traces back several decades, with roots in aerospace engineering and industrial automation. Historically, the first DT was created by NASA in 1970 during Apollo 13 programme to train personnel and troubleshoot for space exploration (Ferko *et al.*, 2022). Over time, the application domains of DTs have expanded, encompassing sectors such as health care (Liu *et al.*, 2019; Pizzolato *et al.*, 2019; Chakshu *et al.*, 2019, 2021; Chakshu and Nithiarasu, 2022; Tamaddon Jahromi *et al.*, 2022), energy Zhou *et al.* (2020); Francisco *et al.* (2020), civil engineering (Kumar *et al.*, 2018; Shim *et al.*, 2019) and manufacturing (Abadías Llamas *et al.*, 2020; Gupta and Basu, 2019). In the area of additive manufacturing, very recent research articles and reviews show that the state of the art is still focusing on reduced order modelling. The obvious final aim is to control, in real-time, the 3D printing process (Afazov *et al.*, 2022; Wei *et al.*, 2021; Li and Polydorides, 2023; Montoya-Zapata *et al.*, 2021).

In the realm of heat transfer, the DT is not always intended as a compound system where there is a continuous data exchange between the physical system and its digital copy, according to the definition reported by Yao *et al.* (2023). Below, we provide a more specific literature survey of DT in the heat transfer field to support the said claim. We provide examples of actual DTs and others that can be categorized as "modelling" and not DT according to the definition used so far.

Yang *et al.* (2023) propose an actual DT to optimize the secondary cooling and final electromagnetic stirring in a continuous casting process. They do that via an offline and then online calibration of a 2D heat transfer model (of an equivalent solid medium) coupled to a neural network (NN) to enhance the accuracy of the (stationary) boundary conditions. In this case, the final aim is to reduce the macro-segregation phenomena in the semi-finished product.

In the work by Nget *et al.* (2023), a multi-physics model of microwave cooking is presented. An optimization to maximize the uniformity of the temperature field inside food is planned as future development in a DT, but no more details are described. Another example of DT is instead shown by Leeming *et al.* (2023). They delve into a DT to effectively

control the supersaturation in batch crystallization. This is an example where no AI/ML/DL algorithm is exploited because the DT is only based on a one-dimensional population balance model.

Chen *et al.* (2023) build a thermodynamic model and an algorithm for solving temperatures in a plate heat exchanger with variable fluid properties. Even though the authors claim to deal with a DT, no strategy allows the digital copy to control, for example, the outlet temperature of the heat exchanger. Similarly, Spateri *et al.* (2023)'s work presented a validated simplified electro-thermal model to manage thermoforming. Even if the model potentially fits well for its very low computational costs (it is a zero-dimensional model) for a real-time implementation in a DT, no real-time control/optimization is proposed.

Cui *et al.* (2023) reproduce a DT of a supercritical coal-fired boiler for detecting temperature anomalies in advance. The DT is fully driven by experimental data coming from temperature sensors and it is based on Particle Swarm Optimization-XGBoost machine learning algorithms. Hachem *et al.* (2021) developed a DRL model to minimize the Nusselt number in 2D/3D fluid-dynamics cases. For example, in a square cavity natural convection case, they vary the distribution of the hot temperature boundary condition keeping the cold one at a constant value. Although the DRL is intrinsically close to a DT concept, the exchange of information between the CFD model (considered as the physical world, the environment in the DRL language) and the NN occurs only during the learning phase. The model is stationary and is aimed at finding the temperature distribution to be applied as a boundary condition in order to minimize the Nusselt number. Similarly, Beintema *et al.* (2020) focus on the same case study, comparing two different control strategies. The first one is the linear-proportional controller, and the second one is based on a DRL model. The results show that the second one is able to achieve a lower Nusselt number. Another similar example of DRL-based heat transfer control is provided by Renault *et al.* (2023). The latter three works (Hachem *et al.*, 2021; Beintema *et al.*, 2020; Renault *et al.*, 2023) share the same learning algorithm: the single step proximal-policy-optimization. In these cases, also known as open-loop control problems, the policy optimization does not depend on the environment response. On the other hand, a closed-loop control of a heat transfer problem is realized by Wang *et al.* (2023). They confirm the DRL-based control strategy, made by an oscillatory flow rate, is the best one to minimize the maximum temperature in the system. Furthermore, even though the authors never mention it, this work may be considered a DT as the NN actions rely on the current state of the environment (CFD simulation).

The above literature review illustrates DT in heat transfer is still a new and emerging research area compared to other sectors, where it is more mature. It emerges that sometimes the concept of DT does not correspond to that one proposed by Yao *et al.* (2023). The other articles, that effectively develop a DT according to the general definition given in this article, build DTs fully driven by experimental data or DTs that deal with stationary or pseudo-stationary heat transfer systems.

The main objective and main novelty over already published papers of this work is to propose a digital twinning of an intrinsically unsteady heat transfer system in a solid medium. More specifically, this work aims at addressing the following question. Given a thermal system with a known transient inward heat flux boundary condition, what is the value of cooling (in terms of heat transfer coefficient boundary condition) we should apply to dynamically control the maximum temperature below a threshold temperature value? In cases where a physical system is not available to provide experimental data, a numerical model can serve as a surrogate system. As demonstrated in a previous work

(Samaniego *et al.* (2020)), this surrogate system can also be a DL model that represents the governing equations of the considered physics, eliminating the need for numerical discretization. However, we assume a finite element method (FEM) model to be our surrogate physical system, while a DNN the digital copy of the system. The DNN model acts as a DT and provides the heat transfer coefficient to meet our goal. Note that this clearly does not represent a reduced order optimization model of the FEM simulation or a simple Inverse Thermal Problem (ITP) aimed at determining a heat transfer coefficient. We couple them, guaranteeing a continuous exchange of data to control the maximum temperature of the system. We adopt a training based on steady-state data to reach our goal. We adopt a training based on steady-state data to reach our goal. Although the results shown later are satisfactory, such results cannot be guaranteed for completely different transient thermal systems.

The rest of the article is organized as follows. Section 2 introduces the DT of the proposed thermal system, Section 3 demonstrates the proposed digital twinning through a number of examples and Section 4 draws some conclusions (Table 1).

2. Digital twin of a thermal system

2.1 The finite element method model – the surrogate physical system: governing equation and dimensionless form

Without a physical system ready to produce continuous experimental data, a FEM model is assumed to be our surrogate physical system. Figure 1(a) shows the basic thermal problem chosen to demonstrate the proposed DT. An unsteady heat flux boundary condition $q(t)$ is applied on the top surface E3. On the opposite bottom boundary, a convective heat transfer boundary condition $h_c(t)$ is assumed with an unknown heat transfer coefficient (E1). It has to be determined by the DT. The left E4 and right E2 vertical surfaces are considered adiabatic. In this case, the problem is reduced to a one-dimensional model. A two-dimensional setup is built by switching the right adiabatic boundary condition with the heat flux condition on the top and vice-versa (see "Step 2. DL model training - Python", left). Finally, a 3D problem is constructed by adopting an isothermal boundary condition on $F6$ as shown in Figure 1(b) and "Step 2. DL model training - Python" (right). The maximum element size used in the calculations is $0.1L$ where L is a characteristic length in the problems of interest. Second-order polynomial functions for discretization, as well as a second-order implicit time scheme with an adaptive time step, are used to carry out all the numerical simulations. The governing equation is the heat conduction in a solid medium, which reads as follows:

Label	1D	2D	Label	3D
E1	Convective	Convective	F1	Convective
E2	Adiabatic	Heat flux	F2	Heat flux
E3	Heat flux	Adiabatic	F3	Adiabatic
E4	Adiabatic	Adiabatic	F4	Adiabatic
			F5	Adiabatic
			F6	Isothermal

Source: By authors

Table 1
Heat conduction through solids, boundary conditions: 1D–2D cases (left) and 3D cases (right)

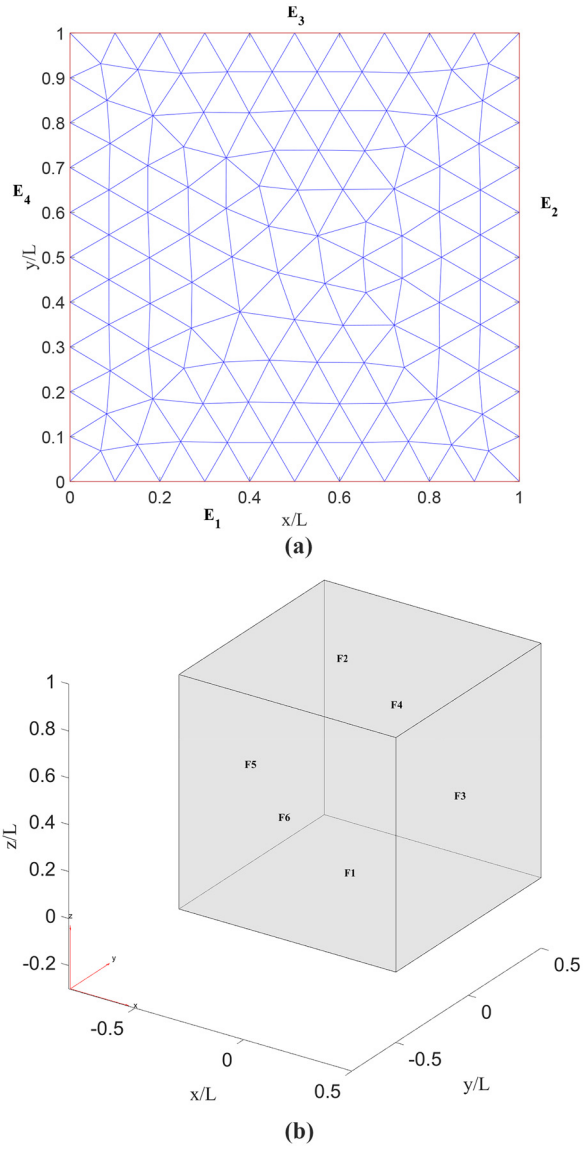


Figure 1.
Heat conduction through solids

Notes: Computational domains. (a) 1D and 2D cases; (b) 3D cases
Source: By authors

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T), \tag{1}$$

where k is the thermal conductivity of the solid medium, ρc its thermal capacity, and their ratio α is the thermal diffusivity. The Neumann and Robin boundary conditions representing the inward heat flux and the cooling convective boundary are described by the following equations:

$$\begin{cases} -k \nabla T \cdot n = q(t), \; on \, \Gamma_q \\ q(t) = h_c(t) \cdot (T - T_0), \; on \, \Gamma_c \end{cases} \tag{2}$$

where Γ_q is the heat flux boundary, Γ_c is the convective boundary and n is the unit normal vector. A dimensionless form of the transient thermal system is provided below. The physical (dimensional) variables are: density ρ, specific heat c, temperature T, time t, thermal conductivity k, heat flux q, heat transfer coefficient h_c and length L. According to the Buckingham theorem (Misic *et al.*, 2010), after a careful evaluation of the relevant physical parameters involved in the problem, as well as the number of fundamental units (kg, m, s, K), we can build the minimum number of dimensionless groups required to describe the unsteady thermal system. The non-dimensional form of the heat equation is:

$$\frac{\partial T^*}{\partial t^*} = \nabla^{2*} T \tag{3}$$

$$\begin{cases} x^* = \dfrac{x}{L} & (\text{dimensionless coordinate}), \\ Fo = t^* = \dfrac{\alpha}{L^2} t & (\text{dimensionless time}), \\ T^* = \dfrac{T - T_0}{T_i - T_0} & (\text{dimensionless temperature}). \end{cases} \tag{4}$$

Non-dimensional quantities of the Neumann boundary conditions are:

$$\begin{cases} q^* = q \dfrac{L}{k(T_i - T_0)} & (\text{dimensionless heat flux}), \\ Bi = \dfrac{h_c L}{k} & (\text{dimensionless heat transfer coefficient}). \end{cases} \tag{5}$$

T_i and T_0 are the initial and the atmospheric temperatures, respectively. The dimensionless time, also known as Fourier number (Fo), takes into account the diffusion and thermal inertia phenomena. In the present work, Fo is varied between 0 and 150. The dimensionless heat flux boundary condition (q^*) is assumed to vary between 0 and 100. Similar to Fo, the Biot number (Bi) is generally introduced in unsteady heat transfer problems to make a convective type boundary condition dimensionless. It is varied between 1 and 15 in the FEM-based data set generation, while it is an unknown (determined by the trained NN) in the full DT as we will see later. Along with these limits, the dimensionless melting temperature is fixed at 100. The FEM model has two different goals. They are:

(1) Generating (stationary) data to train the NN (DL).

(2) Testing the DT (trained DL model) to establish its accuracy in predicting the correct heat transfer coefficient (Biot number) for a given threshold temperature and transient heat flux.

2.2 The deep learning model

As mentioned above, the objective of the DL model is to solve the ITP and obtain the dynamic heat transfer coefficient for a given time-dependent heat flux and threshold temperature. We trained in total three different NNs for 1D, 2D and 3D case studies, respectively. The first two DL models have two inputs (the maximum temperature calculated over the domain and the heat flux applied as a boundary condition) and one output (the heat transfer coefficient). The third one has an additional input as the temperature of the isothermal boundary condition is also varied. The first two DL models have three hidden layers with 32, 8 and 4 neurons respectively while the NN for the 3D cases has an intermediate hidden layer with 16 neurons. The activation function for all inner layers is the hyperbolic tangent (tanh) function. Given that the heat transfer coefficient has no upper bound, a rectified linear unit function ("ReLu") is chosen as output activation function. No particular strategy is used to standardize/normalize the input and output data. The loss function is the classical mean square error (MSE) function. A maximum of 2,000 epochs are used to train the DL models, while 1,064 is the batch size. Adam is the chosen optimizer due to its wide range of applicability, little memory requirements and computational efficiency (Kingma and Ba, 2014). Furthermore, it has been used in other heat transfer DL models as can be seen in the work of Tamaddon-Jahromi *et al.* (2020). To prevent classical problems of ML algorithms such as overfitting/underfitting and local minima, a sensitivity analysis on hyperparameters was carried out to test the robustness of the models. Furthermore, a K-fold cross-validation is implemented, partitioning the data into four sub-sets (or folds). In each iteration, one sub-set is used for validation, and the remaining three are used for training. The process repeats such that each sub-set gets used once as the validation set. The reliability of the model is confirmed by numerical values of root mean square error (RMSE) of both training and validation data sets and their average (for each fold) that stand at the same order of magnitude (see Table 2).

2.3 Finite element method -deep learning coupling (digital twinning)

Assuming we have already trained, validated, and tested the DL (see Section 3), its coupling with the real system (FEM) is essential to control the maximum temperature in real time. The coupling between the FEM (the physical system) and DL is illustrated in Figure 2. Figure 2(a), schematically represents the DT with continuous exchange of data with the FEM model. The DT (the trained DL model) receives time-dependent heat flux and the maximum allowed temperature data and returns a time-dependent heat transfer coefficient to the FEM model. In Figure 2(b) the proposed coupling strategy is presented: the total physical time of the simulation is divided into N intervals. In the last step of k-th interval, the DT is invoked to determine the heat transfer coefficient to be used in the interval $k+1$ of the FEM simulation. The inputs of the DT are the maximum allowed temperature, reduced by a constant safety factor and the maximum applied heat flux in the k-th interval N_k. It is important to remember that, in this article, we train the DL with steady-state data. On the other hand, the real system for the DT is unsteady by definition. Nevertheless, steady-state data can comprehensively establish an effective relationship between heat transfer coefficient, heat flux and maximum temperature. As the heat transfer coefficient (Biot number) calculated by the DT is based on the maximum heat flux recorded during the time

Parameter	NN-1D	NN-2D	NN-3D
Model type	Sequential	Sequential	Sequential
Input features	2	2	3
Layer 1 units	32	32	32
Layer 1 activation	tanh	tanh	tanh
Layer 2 units	8	8	16
Layer 2 activation	tanh	tanh	tanh
Layer 3 units	4	4	8
Layer 3 activation	tanh	tanh	tanh
Layer 4 units	–	–	4
Layer 4 activation	–	–	tanh
Output layer units	1	1	1
Output layer activation	ReLU	ReLU	ReLU
Loss Function	MSE	MSE	MSE
Optimizer	Adam	Adam	Adam
Metric	MAPE, RMSE	MAPE, RMSE	MAPE, RMSE
Epochs	2,000	2,000	2,000
Batch size	1,064 (256 in k-fold)	1,064 (256 in k-fold)	1,064 (256 in k-fold)
Shuffling	Enabled	Enabled	Enabled
Validation split	0.4	0.4	0.4
k-fold c.v., RMSE training-validation	0.056–0.056	0.12–0.14	1.29–1.31
Accuracy	97%	96%	85%

Source: By authors

Table 2. Neural network architectures and hyperparameters used

interval, it is higher than what would be produced by a DT trained on transient data (see Appendix 1). As confirmed by the results in the next section, the proposed approach works effectively. A safety factor is applied to the threshold temperature to take into account the fact that the heat transfer coefficient is not updated instantaneously with heat flux and maximum temperature variations. A further analysis shown in Appendix 1 aims to clarify that the correlation between average heat flux, maximum temperature, and Biot number is the same in steady and transient simulations. The adopted strategy based on steady-state data allows us to drastically reduce the computational costs in both dataset generation and the DT training.

3. Results and discussion

3.1 Steady-state finite element method data for training

As mentioned above, the dimensionless heat flux on the top surface (see Figure 1) is assumed to be between 0 and 100, while the Biot number on the bottom varies between 1 and 15. For every sample (q^*,Bi), the maximum temperature over the domain is calculated. The three variables (dimensionless heat flux, Biot number and maximum dimensionless temperature) are obviously physically related. As seen in Figure 3, for specific heat flux, the higher the Biot number, the lower the maximum temperature and vice-versa. For a specific heat transfer coefficient, an increase in heat flux corresponds to an increase in the maximum temperature. These trends are summarized by contours of the 1D results in Figure 4, where the Biot number and heat flux are represented along the x and y axes, respectively. Examples of 2D and 3D temperature maps are provided in Figure 5. In these simulations, the boundary conditions are specified in the Figure 5 caption.

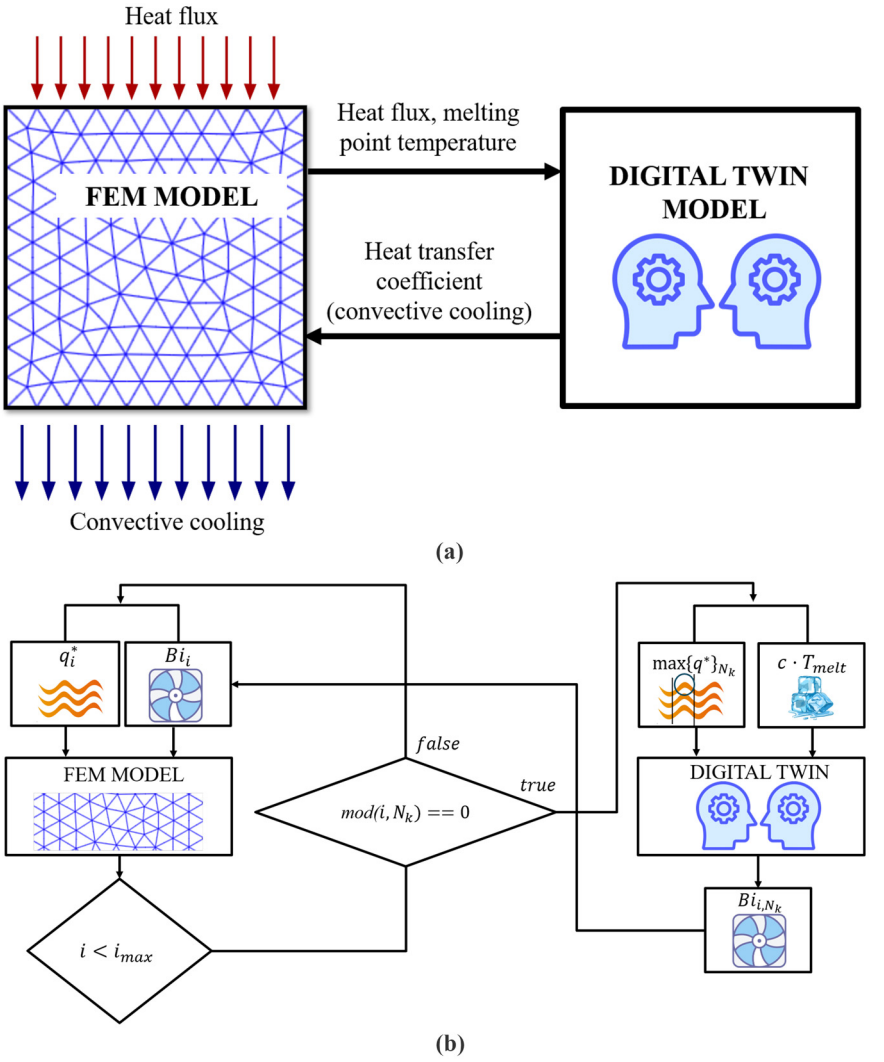
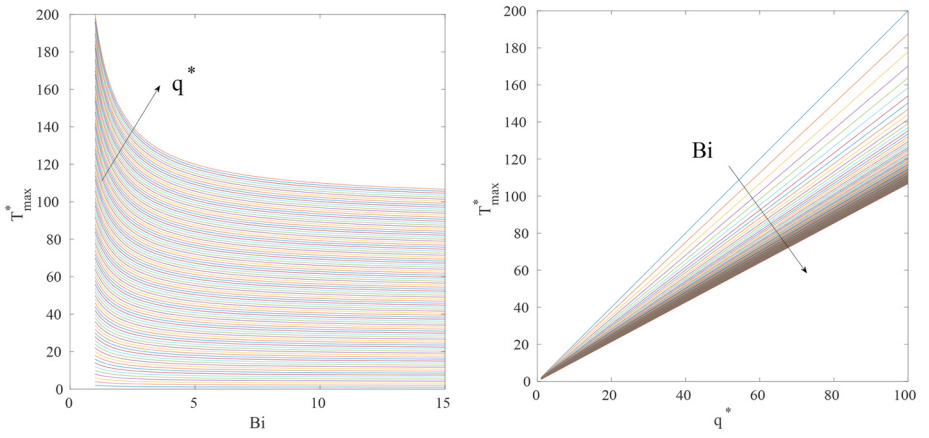


Figure 2.
Digital twinning

Notes: (a) Schematic of FEM and DT coupling; (b) flow chart explaining different steps of digital twinning
Source: By authors

3.2 Digital twin training, validation and testing

The validation-split ratio is 0.4 and the classical mean-squared-error (MSE), equation (6) is chosen as loss function as specified in Table 2. As an additional metric parameter to evaluate the training accuracy, the mean-absolute-percentage-error (MAPE) is calculated equation (7). MSE and MAPE are given as:

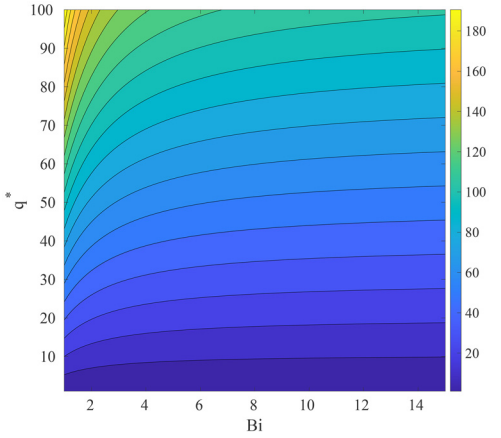


Notes: (a) Maximum temperature vs heat transfer coefficient for different heat fluxes; (b) Maximum temperature vs heat flux for different heat transfer coefficients
Source: By authors

Figure 3.
DL training and validation data

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{6}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100, \tag{7}$$



Note: Contours of the maximum temperature against heat transfer coefficient and heat flux
Source: By authors

Figure 4.
DL training and validation data

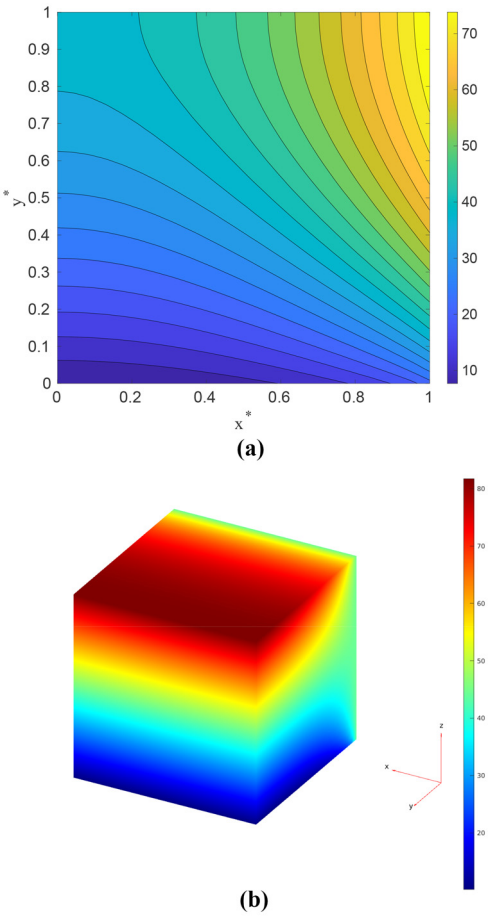


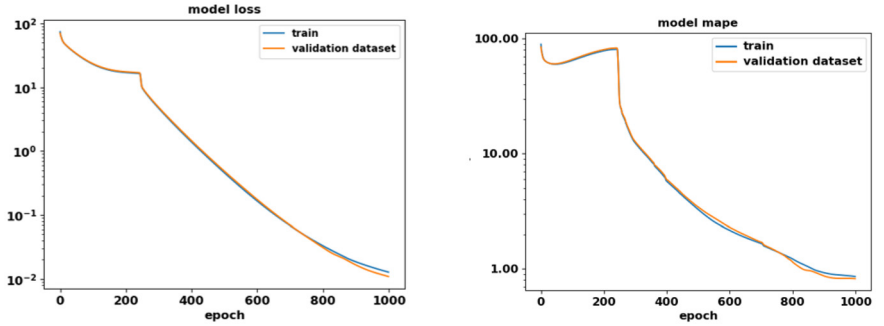
Figure 5.
DL training and validation data: 2 D and 3 D examples

Notes: (a) 2D – $Bi = 7, q^* = 80$;
(b) 3D – $Bi = 7, q^* = 80$ $Bi = 7, q^* = 80$, $T^* = 50$
Source: By authors

where, n is the number of data points in the data set, y_i and $\hat{y}_i$ represent the actual (observed) and the predicted values of the data point i, respectively. An example of MSE and MAPE trends, for both training and validation data sets, against the number of training epochs is pictured in Figure 6. For 1D-2D problems, MSE values reach about 10^{-2} after 1,000 epochs. MAPE values are around 1%. The accuracy of 3D problems is lower. The numerical results of the k-fold cross-validation are reported in Table 2.

The testing is performed on completely new/unseen data adopting the strategy shown by the flowchart in Figure 7.

To evaluate the performance of the DL, we assume that the prediction is "true" when MAPE between the heat transfer coefficient predicted by the DL and that of the one used in the FEM model as boundary condition is less than 2.5%. With this assumption, a user-



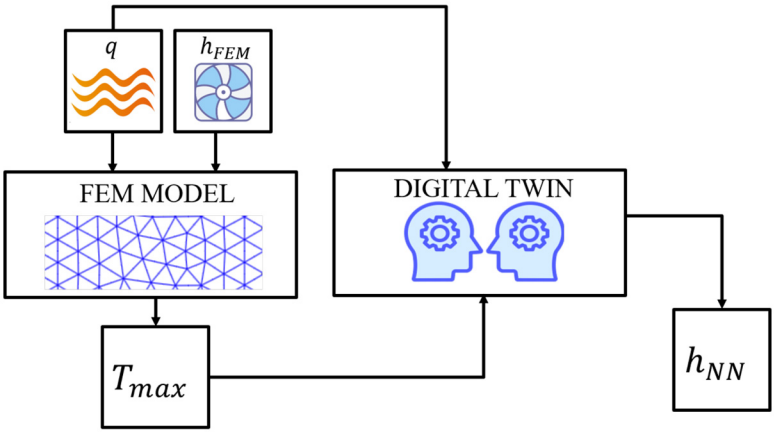
Notes: (a) MSE (loss function) vs epochs; (b) MAPE vs epochs
Source: By authors

Figure 6.
Training the DT

defined accuracy (like for a classification problem) is calculated as the ratio between true predictions and all data (see Figure 8). The confusion matrix shown in Figure 8, is generally adopted for such kinds of problems. "Positive" means that prediction is higher than the real heat transfer coefficient, while "Negative" represents the opposite case. The results are satisfactory because the MAPE is larger than 2.5% only in 21 out of 1,000 samples. Among such samples, only in four the predicted heat transfer coefficient is lower than the real one. The accuracy, calculated as the ratio of the numbers of true/false positive cases over the total cases is 97.9%.

3.3 Digital twinning results and discussion

The digital twinning of the thermal system includes the following steps. In the first step, a comprehensive physics-based data set, covering different physical parameters, is generated. The second step involves training, validation and testing of a DNN to generate a DT of the



Source: By authors

Figure 7.
Flowchart listing the steps in testing the DL-based DT

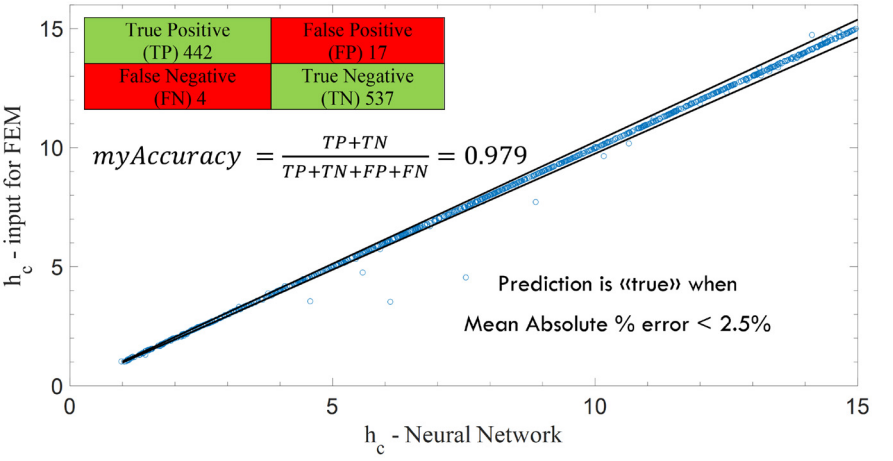


Figure 8.
Testing: real vs predicted heat transfer coefficient, confusion matrix and accuracy

Source: By authors

physical system. Once the DT is trained, validated and tested, it is ready for twinning with the physical system, which is the third and final step. The computer codes, implemented in Matblab/Python, are provided in the Appendix 2. We use a surrogate FEM model in this study to represent the physical system as shown in Figure 2. In the following sub-sections, we provide results from different examples, including linear and non-linear 1D, 2D and 3D problems, to demonstrate the effectiveness of the proposed digital twinning procedure.

3.3.1 Variable dimensionless heat flux. Four different dimensionless heat flux time series are applied on the top boundary (1 D case, see Figure 1 and "Step 2. DL model training - Python"). They are:

- Sinusoidal distribution

$$q_1^*(t^*) = 20\sin(2\pi t^*/30) + 60, \tag{8}$$

- Gaussian pulse

$$q_2^*(t^*) = 45e^{-\left(\frac{t^*-15}{2}\right)^2} + 50, \tag{9}$$

- Compound function with a tan and sin

$$q_3^*(t^*) = 90 \cdot norm[\arctan(t^* - 5) + 0.2\sin(2\pi t^*)], \tag{10}$$

- Linear heat flux

$$q_4^*(t^*) = 3 \cdot t^*. \tag{11}$$

Here, the *norm* operator normalizes a function between 0 and 1. Now we dynamically couple the FEM solution and DT model in time as shown in Figure 2. This results in continuous data exchange between the physical system and its DT. The physical system provides time-dependent heat flux applied on it and the threshold temperature (e.g. material failure or melting point temperature) to the DT. The DT returns an appropriate, time-dependent heat transfer coefficient that restricts the maximum physical system temperature just below the threshold maximum temperature. Since there could be a time lag between the time-dependent heat flux applied and the evolution of the maximum temperature that depends on the transient response, we may need a safety factor to reduce the threshold temperature supplied to the DT. We fix it at 0.95 for all simulations.

The results are shown in Figures 9(a)–(c), respectively, for the applied heat fluxes of equations (8), (9) and (10). It is apparent that in all the cases the maximum temperature is below the melting temperature and the system is safe. In the first interval, the heat transfer coefficient is initialized with a maximum value of 15. Then, after every interval the DT tries to find the minimum heat transfer coefficient to guarantee a temperature below the maximum temperature constraint, but at the same time, very close to the maximum allowed temperature. It is obvious but essential to specify that it is practically impossible to obtain a constant maximum temperature if the boundary conditions are unsteady.

In Figure 9(a), we can see how the heat transfer coefficient sharply decreases from the maximum value and then follows the trend of the applied sinusoidal heat flux. As seen, the maximum temperature starts from the zero initial condition and increases until it reaches a value close to but below the maximum allowed temperature. Later, the dimensionless heat transfer coefficient flattens, reaching a stable value of unity (minimum value of the Biot number allowed). As observed in Figure 3(a), increasing the heat transfer coefficient above a certain value will have little effect on the maximum temperature for a given heat flux value. On the other hand, the heat transfer coefficient cannot be zero to avoid an infinite value of temperature. Therefore, the maximum and minimum heat transfer coefficient limits are imposed by physics, not by the DT implementation or the NN. For this reason, they are carefully taken into account when generating data, as well as in the testing and the DT architecture.

In Figure 9(a), the heat transfer coefficient initially drops sharply but gains momentum to follow the heat flux variation. When the applied heat flux drops during the latter part of the calculation, the heat transfer coefficient flattens to the minimum value allowed (unity). In Figure 9(b), we can see that, before and after the pulse, the heat transfer coefficient is constant (minimum allowed value) as the heat flux is at its pseudo-stationary state. The heat flux pulse clearly produces an increase in the maximum temperature. The heat transfer coefficient follows the same pattern, even with a low sampling frequency. We choose, as also in the other cases, to split the total 900 time steps of the simulations into 100 intervals. It means that the heat transfer coefficient is a piece-wise function, changing its value every nine time steps. The third case, shown in Figure 9(c), presents a heat flux that is steeply increasing from zero to about 90 with high-frequency oscillation given by the sinusoidal component. Even in this case, the heat transfer coefficient starts from its maximum value. Then, it reaches a plateau as heat flux is very low and then goes up with an obvious delay with respect to heat flux. In the last case shown in Figure 9(d), a linear heat flux is assumed. It results that when Fo is less than 50, the Biot number stands at its minimum allowed value while the maximum temperature linearly increases. Later, a non-linear increase in the Biot number determined by the trained DT enables to reach an approximately constant maximum temperature.

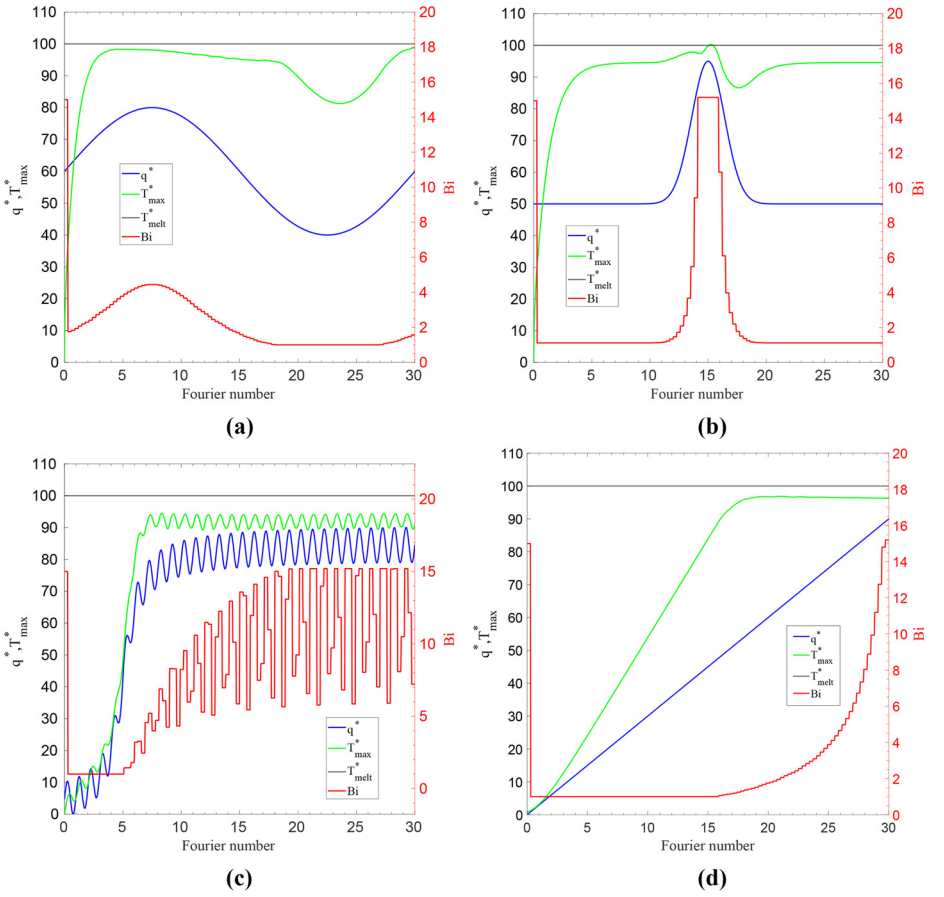
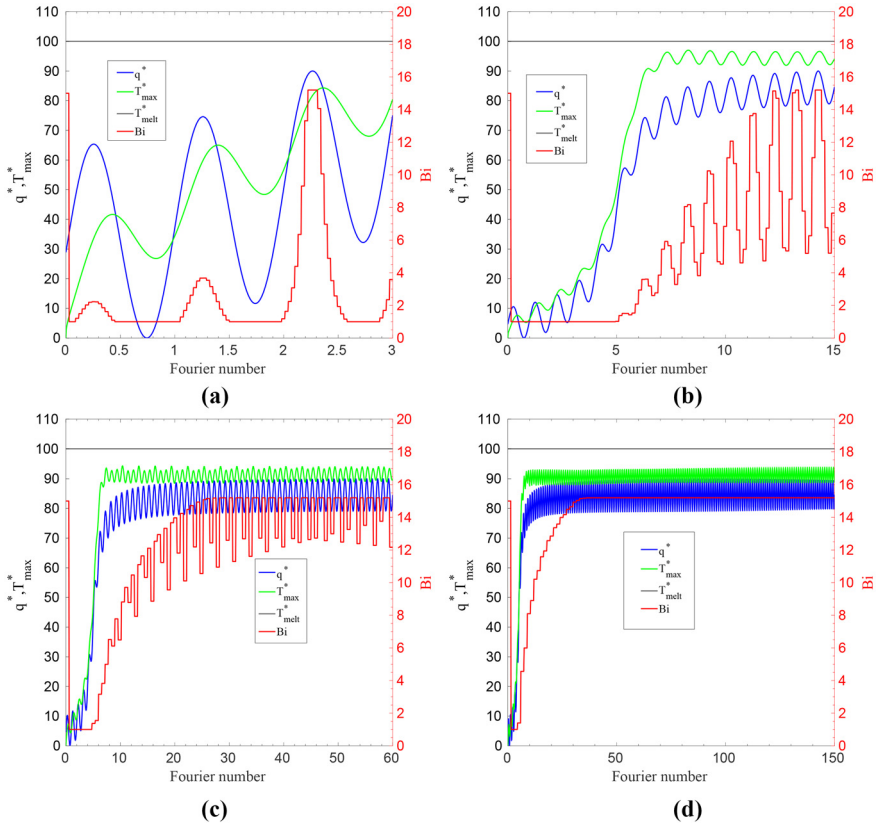


Figure 9.
Digital twinning of a 1D heat transfer system

Notes: Linear heat transfer with heat flux boundary condition given in (a) equation (8); (b) equation (9); (c) equation (10); (d) equation (11)
Source: By authors

*3.3.2 Variable thermal diffusivity (*Fo*).* The objective of this sub-section is to demonstrate that the DT, being trained on dimensionless data, effectively works also with different physical thermal diffusivity and therefore different transient responses of the thermal system. In practical terms, the physical time window is fixed and different thermal properties are chosen by varying the Fourier number. We adopt equation (10) as test case. Four different Fo are simulated $Fo = 3, 15, 60, 150$. A low Fo indicates a stronger internal energy stored compared to conductive transport and the opposite happens at higher Fo. The results are shown in Figure 10. Figure 10(a) shows the results for lower Fo while results for the highest Fo are illustrated in Figure 10(d). At the lowest Fo numbers, the time shift between boundary conditions and maximum temperature indicates a higher thermal inertia and a slower thermal transport compared to the other three cases. From case (b) and (c) in Figure 10, the maximum temperature profile becomes more noisy as the thermal inertia of



Notes: Varying the Fourier number: (a) $Fo = 3$; (b) $Fo = 15$; (c) $Fo = 60$; (d) $Fo = 150$
Source: By authors

Figure 10.
Digital twinning of a 1D heat transfer system

the system reduces with an increase in *Fo*. The Biot number variation still follows the heat flux variations even if the profiles are more "discrete" compared to the case in Figure 10(a). In the last case, pictured in Figure 10(d), the Biot number remains constant after $Fo = 30$, as the DT always sees the same value of (maximum) heat flux as input into the DT.

3.3.3 Digital twinning of a non-linear case: temperature-dependent thermal conductivity. Here the reliability of the same DT for a non-linear heat transfer case with a temperature-dependent thermal conductivity is investigated. The analytical expression of our choice takes into account the previous physical limitations. We consider the following formula to express non-linearity:

$$k(T) = 1 + \frac{T}{T_{melt}} \tag{12}$$

where T_{melt} is the maximum allowed temperature. Figure 11(a) shows the results obtained using equation (10). As seen, the heat transfer coefficient required to have a maximum

temperature below the maximum allowed temperature is smaller than the linear case (see Figure 9) due to the overall higher thermal conductivity.

3.3.4 2D case. As mentioned above, to make the problem two-dimensional, it is sufficient to switch the heat flux and right side adiabatic boundary condition [see Figure 1(a)]. In this case, the number of features and labels correspond to those of the NN used in the previous DT. Also, the retraining of the NN is done starting from the 1D model to accelerate the process. Furthermore, in this case, using a safety factor equal to 0.95, the DT is able to maintain the maximum temperature below the melting point, as shown in Figure 11.

3.3.5 3D cases. The 3 D cases are the final cases presented in this work. The additional isothermal boundary condition included in the computational domain (see "Step 2. DL model training – Python") is considered an added feature to clearly improve the accuracy of the NN. The dimensionless temperature value on the isothermal boundary varies between 0 and 70. The higher the temperature the higher will be the Bi number to maintain the maximum temperature under control. Two cases are presented in Figure 12 with isothermal boundary values of 50 [Figure 12(a)] and 70 [Figure 12(b)]. It can be noted in both cases that the maximum temperature location changes with time [Figure 12(c)]. At the beginning of the simulations, as the initial condition T_i^* is fixed at 1, the maximum temperature is located closer to the isothermal boundary (F2, 1 (b)). Later on, the location of the maximum temperature moves towards the heat flux boundary (F2), when Fo is approximately 5. To prove that, the dynamic position of maximum temperature for the case shown in Figure 12(a), in terms of the three dimensionless coordinates $x^*_{max}, y^*_{max}, z^*_{max}$, is pictured in Figure 12(c). For easier reading of Figure 12(c), it is important to remember that the plane equations where the isothermal (F6) and heat flux boundary (F2) conditions [Figure 1(b)] were applied are $y^* = 0$ and $z^* = 1$, respectively. Nevertheless, the 3D proposed digital twinning approach maintains the maximum temperature below the threshold temperature assumed.

4. Conclusions

In this article, a digital twinning approach to thermal systems was presented. The literature review demonstrated how digital twinning is still an emerging topic in this field. The work

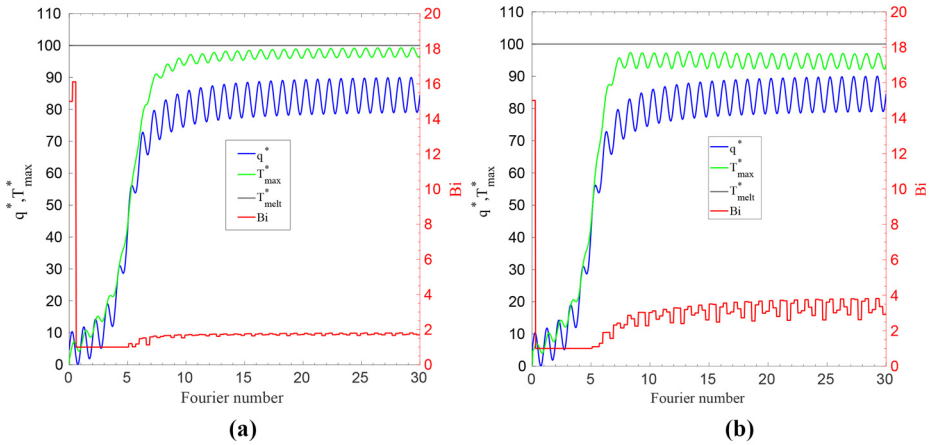
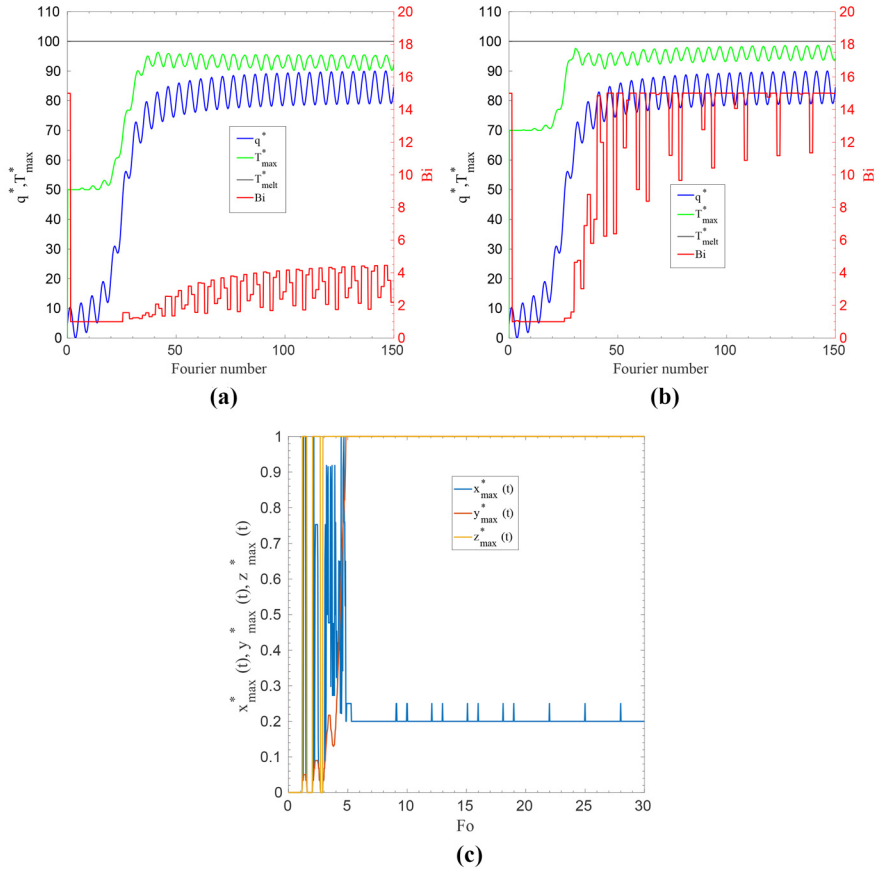


Figure 11.
Digital twinning of non-linear (1D) and two-dimensional cases

Notes: (a) Non-linear heat transfer case; (b) 2D case
Source: By authors



Notes: (a)T^*_{iso} = 50; (b) T^*_{iso} = 70; (c) dynamic location of maximum temperature (T^*_{iso} = 50)

Source: By authors

Figure 12.
1D cases without safety factor

presented here is therefore among the first to focus on a digital twinning approach with a clear aim to control a transient thermal system. The digital twinning approach here means a digital copy of a real system able to exchange information continuously with the physical world and influence each other. Although we assumed a surrogate FEM model for the physical system, demonstrating such a digital twinning process on a real physical system is the ultimate aim and the results presented here are the first step towards such an aim.

The objective of the digital twinning process in the presented work was to control the maximum temperature inside the system in the presence of an unsteady heat input into the system. However, the digital twin training was based on stationary data. The goal of the digital twinning process was to limit the maximum temperature below the maximum allowed temperature by allowing the DT to choose an appropriate convective heat transfer coefficient. Once the relevant and independent dimensionless variables were identified

through a systemic dimensional analysis, a sensitivity analysis based on their variations was carried out. Furthermore, 1 D, 2 D and 3 D FEM setups were investigated to test the DT. The training of the DL model was made separately for the three cases. The results clearly showed that limiting the maximum temperature below the maximum allowed temperature was possible even in the two and three-dimensional cases. In all cases, a safety factor was applied on the threshold maximum temperature before being passed on to the DT. The limitations of the proposed work include a static DT, trained on steady-state data. Even if the results shown in the present article demonstrate that the DT works effectively to control the maximum temperature of the system, similar results in completely different systems (in terms of dimensionless variables, their ranges, and boundary conditions) cannot be established a priori. Another limitation is the lack of availability of a real physical system. In conclusion, further studies should be aimed at developing a continuously changing DT, via a reinforcement learning platform to rapidly represent any new data.

References

Abadías Llamas, A., Bartie, N., Heibeck, M., Stelter, M. and Reuter, M.A. (2020), "Simulation-based exergy analysis of large circular economy systems: zinc production coupled to CDTE photovoltaic module life cycle", *Journal of Sustainable Metallurgy*, Vol. 6 No. 1, pp. 34-67.

Afazov, S., Roberts, A., Wright, L., Jadhav, P., Holloway, A., Basoalto, H., Milne, K. and Brierley, N. (2022), "Metal powder bed fusion process chains: an overview of modelling techniques", *Progress in Additive Manufacturing*, Vol. 7 No. 2, pp. 289-314, doi: 10.1007/s40964-021-00230-1.

Aldaghi, A., Banejad, A., Kalani, H., Sardarabadi, M. and Passandideh-Fard, M. (2023), "An experimental study integrated with prediction using deep learning method for active/passive cooling of a modified heat sink", *Applied Thermal Engineering*, Vol. 221, p. 119522, doi: 10.1016/j.applthermaleng.2022.119522.

Beintema, G., Corbetta, A., Biferale, L. and Toschi, F. (2020), "Controlling Rayleigh–Bé nard convection via reinforcement learning", *Journal of Turbulence*, Vol. 21 Nos 9/10, pp. 585-605, doi: 10.1080%2F14685248.2020.1797059.

Chakshu, N.K. and Nithiarasu, P. (2022), "An ai based digital-twin for prioritising pneumonia patient treatment", *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, Vol. 236 No. 11, pp. 1662-1674, doi: 10.1177/09544119221123431.

Chakshu, N.K., Sazonov, I. and Nithiarasu, P. (2021), "Towards enabling a cardiovascular digital twin for human systemic circulation using inverse analysis", *Biomechanics and Modeling in Mechanobiology*, Vol. 20 No. 2, pp. 449-465, doi: 10.1007/s10237-020-01393-6.

Chakshu, N.K., Carson, J., Sazonov, I. and Nithiarasu, P. (2019), "A semi-active human digital twin model for detecting severity of carotid stenoses from head vibration-a coupled computational mechanics and computer vision method", *International Journal for Numerical Methods in Biomedical Engineering*, Vol. 35 No. 5, p. e3180, doi: 10.1002/cnm.3180, available at: www.scopus.com/inward/record.uri?eid=2-s2.0-85061823271&doi=10.1002%2fcnm.3180&partnerID=40&md5=29979cd77724f0b1334fc64faf9e3a09

Chen, L., Zhao, K. and Tao, W.Q. (2023), "Research on one-dimensional digital twin algorithm of plate heat exchanger", *Numerical Heat Transfer; Part A: Applications*, pp. 1-20, available at: www.tandfonline.com/doi/full/10.1080/10407782.2023.2222906

Cui, Z., Xu, J., Liu, W., Zhao, G. and Ma, S. (2023), "Data-driven modeling-based digital twin of supercritical coal-fired boiler for metal temperature anomaly detection", *Energy*, Vol. 278, p. 127959, doi: 10.1016/j.energy.2023.127959.

Ferko, E., Bucaioni, A. and Behnam, M. (2022), "Architecting digital twins", *IEEE Access*, Vol. 10, pp. 50335-50350.

Francisco, A., Mohammadi, N. and Taylor, J.E. (2020), "Smart city digital twin–enabled energy management: toward real-time urban building energy benchmarking", *Journal of Management in Engineering*, Vol. 36 No. 2, p. 4019045.

Grabe, C., Jäckel, F., Khurana, P. and Dwight, R.P. (2023), "Data-driven augmentation of a rans turbulence model for transonic flow prediction", *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 33 No. 4, pp. 1544-1561, doi: 10.1108/HFF-08-2022-0488.

Guo, Z., Wang, Y., Zhao, S., Zhao, T. and Ni, M. (2023), "Modeling and optimization of micro heat pipe cooling battery thermal management system via deep learning and multi-objective genetic algorithms", *International Journal of Heat and Mass Transfer*, Vol. 207, p. 124024, doi: 10.1016/j.ijheatmasstransfer.2023.124024.

Gupta, A. and Basu, B. (2019), "Sustainable primary aluminium production: technology status and future opportunities", *Transactions of the Indian Institute of Metals*, Vol. 72 No. 8, pp. 2135-2150.

Hachem, E., Ghraieb, H., Viquerat, J., Larcher, A. and Meliga, P. (2021), "Deep reinforcement learning for the control of conjugate heat transfer", *Journal of Computational Physics*, Vol. 436, p. 110317, doi: 10.1016/j.jcp.2021.110317.

Keramati, H., Hamdullahpur, F. and Barzegari, M. (2022), "Deep reinforcement learning for heat exchanger shape optimization", *International Journal of Heat and Mass Transfer*, Vol. 194, p. 123112, doi: 10.1016/j.ijheatmasstransfer.2022.123112.

Kingma, D.P. and Ba, J. (2014), "Adam: a method for stochastic optimization", arXiv preprint arXiv:1412.6980.

Kumar, S.A., Madhumathi, R., Chelliah, P.R., Tao, L. and Wang, S. (2018), "A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance", *Journal of Reliable Intelligent Environments*, Vol. 4 No. 4, pp. 199-209.

Leeming, R., Mahmud, T., Roberts, K.J., George, N., Webb, J., Simone, E. and Brown, C.J. (2023), "Development of a digital twin for the prediction and control of supersaturation during batch cooling crystallization", *Industrial and Engineering Chemistry Research*, Vol. 62 No. 28, pp. 11067-11081, doi: 10.1021/acs.iecr.3c00371.

Li, X. and Polydorides, N. (2023), "Fast heat transfer simulation for laser powder bed fusion", *Computer Methods in Applied Mechanics and Engineering*, Vol. 412, p. 116107, doi: 10.1016/j.cma.2023.116107.

Lin, Q., Hong, J., Liu, Z., Li, B. and Wang, J. (2018), "Investigation into the topology optimization for conductive heat transfer based on deep learning approach", *International Communications in Heat and Mass Transfer*, Vol. 97, pp. 103-109, doi: 10.1016/j.icheatmasstransfer.2018.07.001.

Liu, Y., Zhang, L., Yang, Y., Zhou, L., Ren, L., Wang, F., Liu, R., Pang, Z. and Deen, M.J. (2019), "A novel cloud-based framework for the elderly healthcare services using digital twin", *IEEE Access*, Vol. 7, pp. 49088-49101.

Löhner, R., Antil, H., Tamaddon-Jahromi, H., Chakshu, N.K. and Nithiarasu, P. (2020), "Deep learning or interpolation for inverse modelling of heat and fluid flow problems?", *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 31 No. 9, pp. 3036-3046, doi: 10.1108/HFF-11-2020-0684.

Misic, T., Najdanovic-Lukic, M. and Nesic, L. (2010), "Dimensional analysis in physics and the buckingham theorem", *European Journal of Physics*, Vol. 31 No. 4, p. 893.

Montoya-Zapata, D., Rodríguez, J.M., Moreno, A., Posada, J. and Ruiz-Salguero, O. (2021), "2d linear finite element simulation of laser metal heating for digital twins", *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 12, p. 11, doi: 10.1051/smdo/2021011.

Nget, S., Mith, H., Boué, G., Curet, S. and Boillereaux, L. (2023), "The development of a digital twin to improve the quality and safety issues of Cambodian pâté: the application of 915 MHZ microwave cooking", *Foods*, Vol. 12 No. 6, p. 1187, doi: 10.3390/foods12061187.

Ongsulee, P. (2017), "Artificial intelligence, machine learning and deep learning", 2017 15th international conference on ICT and knowledge engineering (ICT&KE), IEEE, pp. 1-6.

Park, B.K. and Kim, C.J. (2023), "Unsteady heat flux measurement and predictions using long short-term memory networks", *Buildings*, Vol. 13 No. 3, p. 707, doi: 10.3390/buildings13030707.

Pizzolato, C., Saxby, D.J., Palipana, D., Diamond, L.E., Barrett, R.S., Teng, Y.D. and Lloyd, D.G. (2019), "Neuromusculoskeletal modeling-based prostheses for recovery after spinal cord injury", *Frontiers in Neurorobotics*, Vol. 13, p. 97.

Renault, M., Viquerat, J., Meliga, P., Grandin, G.A., Meynet, N. and Hachem, E. (2023), "Investigating gas furnace control practices with reinforcement learning", *International Journal of Heat and Mass Transfer*, Vol. 209, p. 124147, doi: 10.1016/j.ijheatmasstransfer.2023.124147.

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V., Guo, H., Hamdia, K., Zhuang, X. and Rabczuk, T. (2020), "An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications", *Computer Methods in Applied Mechanics and Engineering*, Vol. 362, p. 112790, doi: 10.1016/j.cma.2019.112790, available at: www.sciencedirect.com/science/article/pii/S0045782519306826

Shim, C.S., Dang, N.S., Lon, S. and Jeon, C.H. (2019), "Development of a bridge maintenance system for prestressed concrete bridges using 3d digital twin model", *Structure and Infrastructure Engineering*, Vol. 15 No. 10, pp. 1319-1332.

Spateri, E., Ruiz, F. and Gruosso, G. (2023), "An electrothermal digital twin for design and management of radiation heating in industrial processes", *IEEE Transactions on Industry Applications*, Vol. 59 No. 5, pp. 1-8, doi: 10.1109/TIA.2023.3287817.

Tamaddon-Jahromi, H.R., Chakshu, N.K., Sazonov, I., Evans, L.M., Thomas, H. and Nithiarasu, P. (2020), "Data-driven inverse modelling through neural network (deep learning) and computational heat transfer", *Computer Methods in Applied Mechanics and Engineering*, Vol. 369, p. 113217, doi: 10.1016/j.cma.2020.113217.

Tamaddon Jahromi, H.R., Sazonov, I., Jones, J., Coccarelli, A., Rolland, S., Chakshu, N.K., Thomas, H. and Nithiarasu, P. (2022), "Predicting the airborne microbial transmission via human breath particles using a gated recurrent units neural network", *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 32 No. 9, pp. 2964-2981, doi: 10.1108/HFF-07-2021-0498, available at: www.scopus.com/inward/record.uri?eid=2-s2.0-85122543576&doi=10.1108%2fHFF-07-2021-0498&partnerID=40&md5=8fab9e7cb148a5c1c3db0ff915edcf93

Wang, Y.Z., He, X.J., Hua, Y., Chen, Z.H., Wu, W.T. and Zhou, Z.F. (2023), "Closed-loop forced heat convection control using deep reinforcement learning", *International Journal of Heat and Mass Transfer*, Vol. 202, p. 123655, doi: 10.1016/j.ijheatmasstransfer.2022.123655.

Wei, H., Mukherjee, T., Zhang, W., Zuback, J., Knapp, G., De, A. and DebRoy, T. (2021), "Mechanistic models for additive manufacturing of metallic components", *Progress in Materials Science*, Vol. 116, p. 100703, doi: 10.1016/j.pmatsci.2020.100703.

Xuereb Conti, Z., Choudhary, R. and Magri, L. (2023), "A physics-based domain adaptation framework for modeling and forecasting building energy systems", *Data-Centric Engineering*, Vol. 4, p. e10, doi: 10.1017/dce.2023.8.

Yang, J., Ji, Z., Liu, W. and Xie, Z. (2023), "Digital-twin-based coordinated optimal control for steel continuous casting process", *Metals*, Vol. 13 No. 4, p. 816, doi: 10.3390/met13040816.

Yao, J.F., Yang, Y., Wang, X.C. and Zhang, X.P. (2023), "Systematic review of digital twin technology and applications", *Visual Computing for Industry, Biomedicine, and Art*, Vol. 6 No. 1, p. 10, doi: 10.1186/s42492-023-00137-4.

Zeeshan, A., Khan, M.I., Ellahi, R. and Asghar, Z. (2023), "Artificial neural network simulation and sensitivity analysis for optimal thermal transport of magnetic viscous fluid over shrinking wedge via RSM", *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 33 No. 10, pp. 3492-3518, doi: 10.1108/HFF-03-2023-0135.

Zhang, Y., Gong, Z., Zhou, W., Zhao, X., Zheng, X. and Yao, W. (2023), "Multi-fidelity surrogate modeling for temperature field prediction using deep convolution neural network", *Engineering Applications of Artificial Intelligence*, Vol. 123, p. 106354, doi: 10.1016/j.engappai.2023.106354.

Zhou, M., Feng, D., Yan, J. and Zhou, X. (2020), "A software platform for second-order responsiveness power grid online analysis", *Power System Technology*, Vol. 44, pp. 3474-3480.

Zhu, F., Chen, J., Ren, D. and Han, Y. (2023), "Transient temperature fields of the tank vehicle with various parameters using deep learning method", *Applied Thermal Engineering*, Vol. 230, p. 120697, doi: 10.1016/j.applthermaleng.2023.120697.

Appendix 1

The objective of this appendix is to prove that the steady-state approach provides a slightly higher maximum temperature compared to a transient simulation. This is done by performing a transient simulation with an unsteady heat flux boundary condition (q_3^*) and a constant Biot number. Similarly, the maximum of the previous transient heat flux, consistently with the FEM-DL coupling adopted in the proposed digital twinning approach, and the same Biot number represent the boundary conditions for the steady-state simulation. Three different combinations of heat flux (by scaling q_3^*) and Biot numbers are tested. The results shown in Figure A1 demonstrate that the correlation between maximum temperature, Biot number and average heat flux are the same in steady and transient simulations. Figure A2 shows three examples of digital twinning without the use of safety factor. As seen, the maximum temperature determined by DT is very close to the threshold temperature. As the temperature occasionally exceeds the threshold temperature, a safety factor slightly below unity is useful.

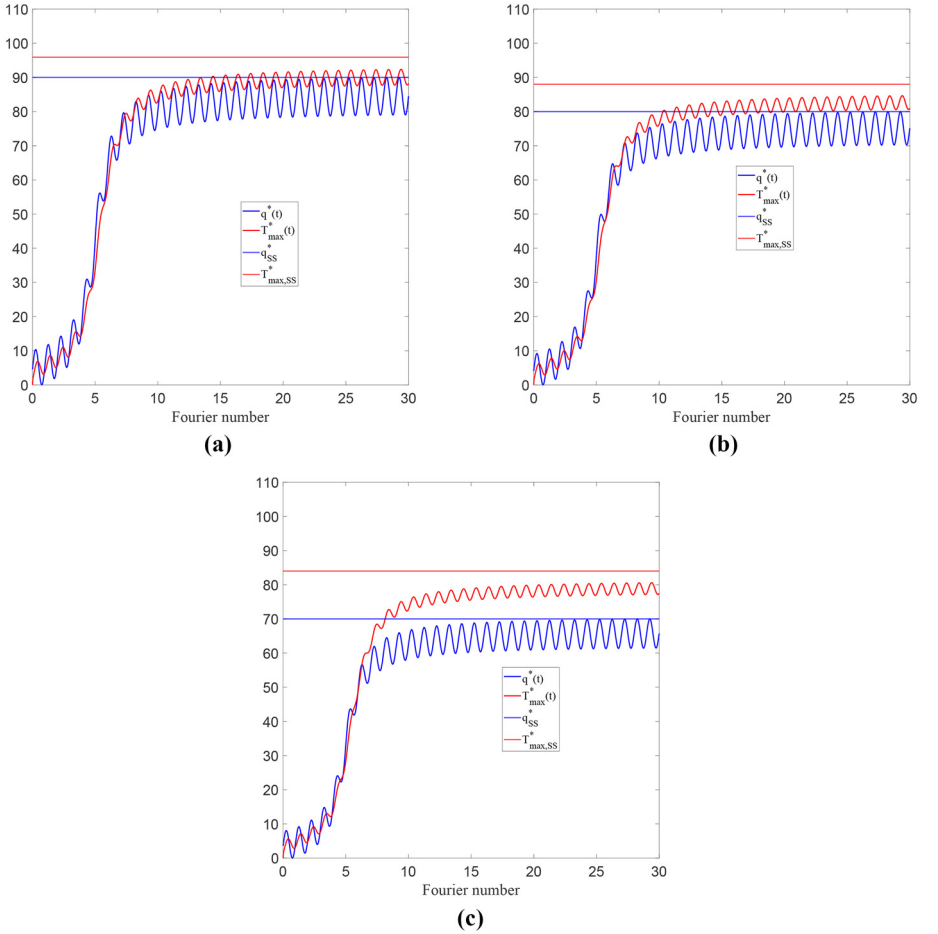
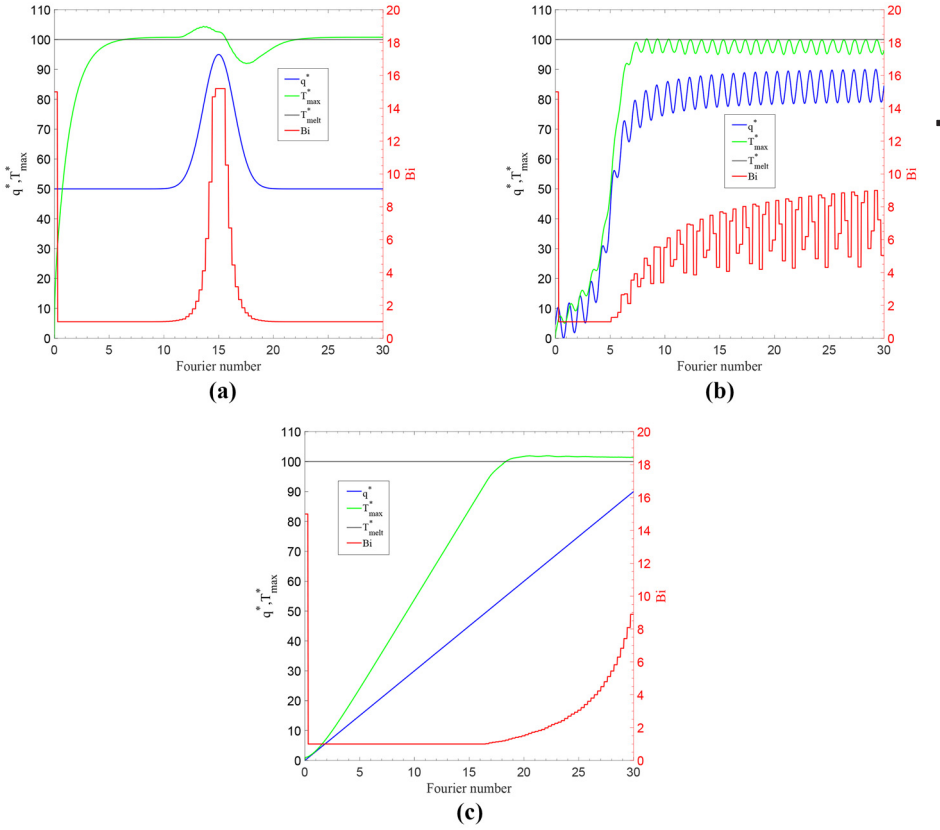


Figure A1.
Transient vs steady-state approach

Notes: $q^*_{max} = 90$, $B1 = 15$; (a)$q^*_{max} = 80$, $B1 = 10$; (b) $q^*_{max} = 70$, $B1 = 5$
Source: By authors

Appendix 2



Notes: (a) Equation (9); (b) equation (10); (c) equation (11)
Source: By authors

Figure A2.
1D cases without safety factor

Step 1: FEM data generation – Matlab

FEM data for cases shown in Figure 1(a)

```
clc
close all
clear
numWorkers = 4;
parpool(numWorkers);
N = 100;
q = rand(1,N^2)*100;
Hmax = 0.1;
Tmax = zeros(N^2,1);
h = rand(1,N^2)*(15-1)+1;

parfor idx = 1:(N*N)

 model = createpde("thermal");
 g = [3 4 0 1 1 0 0 0 1 1]';
 geometryFromEdges(model, decsg(g));
 thermalProperties(model, "ThermalConductivity",1);
 generateMesh(model, 'Hmax', Hmax);
 model.SolverOptions.RelativeTolerance = 1.0e-4;
 model.SolverOptions.AbsoluteTolerance = 1.0e-4;
 thermalBC(model, 'Edge',3,'HeatFlux',q(idx));
 thermalBC(model,"Edge",1,"ConvectionCoefficient",...
 h(idx), "AmbientTemperature",T0);
 results = solve(model);
 Tmax(idx) = max(results.Temperature);

end
delete(gcp('nocreate'));

data = [q',h',Tmax,Tiso];
header = ["q", "h", "Tmax","Tiso"];
name_file = sprintf("steadyData2D_pure.dat");
writematrix(header, name_file, 'Delimiter', '\t')
 writematrix(data, name_file, 'Delimiter', '\t', 'WriteMode',
'append')
```

Source: By authors

Step 2. DL model training – Python

DL model training for cases shown in Figure 1(a)

```
import numpy as np
from keras.layers import Dense, Activation, Flatten, Dropout
from keras.models import Sequential, load_model
from sklearn.model_selection import train_test_split

Nsample = 9000
Nfeatures = 2
```

```
data = np.loadtxt("steadyData.dat", delimiter='\t', skiprows = 1)
print(data.shape)
q  = data[:,0]
h  = data[:,1]
Tmax = data[:,2]

Input_array = np.zeros((Nsample,Nfeatures))
Input_array[:,0] = q
Input_array[:,1] = Tmax
Output_array  = h

 X_train, X_val, Y_train, Y_val = train_test_split(Input_array,
Output_array, test_size = 0.4, shuffle=True)

model = Sequential()
model.add(Dense(32, activation="tanh"))
model.add(Dense(8, activation="tanh"))
model.add(Dense(4, activation="tanh"))
model.add(Dense(1, activation="relu"))
 model.compile (loss="mean_squared_error",optimizer="adam", met-
rics=["mape"])
 history = model.fit(X_train, Y_train, epochs = 2000, batch_size =
1064, validation_data=(X_val, Y_val), verbose = 1, shuffle=True)
```

Source: By authors

Step 3. The digital twinning: FEM–DL coupling – Matlab

```
FEM data for cases shown in Figure 1(a)
 steadymodel = importKerasNetwork('steady1.h5');

 t0  = 0;
 tf  = 30;
 Nt  = 900;
 Nint = 100;
 Nt_int = Nt/Nint;
 t  = linspace(t0, tf, Nt);
 Hmax = 0.1;
 T0  = 0;
 Ti  = 1;
 Tmax = zeros(Nt,1);
 Tmelt = 100;
 hmin = 1;
 h  = ones(Nt,1)*15;
 model = createpde("thermal", "transient");
 g  = [3 4 0 1 1 0 0 0 1 1]';
 geometryFromEdges(model, decsg(g));
 rho = 1;
 c = 1;
 k = 1;
 Fo = k/rho/c/1^2*t;
 q = atan(Fo-5)+0.2*sin(2*pi*Fo);
```

```
q = 90*(q-min(q))/(max(q)-min(q));

                thermalProperties(model,"ThermalConductivity",
k,"MassDensity",1,"SpecificHeat",1);
generateMesh(model, 'Hmax', Hmax);
model.SolverOptions.RelativeTolerance = 1.0e-4;
model.SolverOptions.AbsoluteTolerance = 1.0e-4;
thermalIC(model,Ti);
thermalBC(model,'Edge',3,'HeatFlux',q(1));

        thermalBC(model,"Edge",1,"ConvectionCoefficient",h(1),
"AmbientTemperature",0);
results = solve(model, Fo(1:2));
[maxT,loc0] = max(results.Temperature);
Tmax(2) = maxT(2);
locations = model.Mesh.Nodes;

 for i = 2:Nt-1

  thermalIC(model,results);
  thermalBC(model,'Edge',3,'HeatFlux',q(i));
        thermalBC(model,"Edge",1,"ConvectionCoefficient",h(i),
"AmbientTemperature",T0);
  results = solve(model, Fo(i:i+1));
  [Tmax1,loc] = max(results.Temperature);
  Tmax(i+1) = Tmax1(2);

  if(mod(i,Nt_int)==0)
  kk = k_values(loc(2),i-Nt_int+1:i);
  x = [max(q(i:i+Nt_int-1)),0.95*Tmelt];
  h(i+1:i+Nt_int) = predict(steadymodel,x);

   if(min(h(i+1:i+Nt_int))<hmin)
    h(i+1:i+Nt_int) = hmin;
   end
  end
end
```

Corresponding author
Armando Di Meglio can be contacted at: armando.dimeglio001@studenti.uniparthenope.it