Materials & Design 231 (2023) 112034

Contents lists available at ScienceDirect

Materials & Design

journal homepage: www.elsevier.com/locate/matdes

Physics-informed neural networks for data-free surrogate modelling and engineering optimization – An example from composite manufacturing

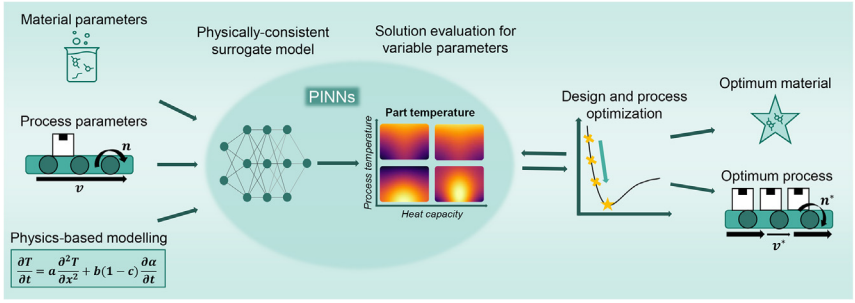
Tobias Würth *, Constantin Krauß, Clemens Zimmerling, Luise Kärger

Karlsruhe Institute of Technology (KIT), Institute of Vehicle System Technology, Karlsruhe, Germany

HIGHLIGHTS

- Physically consistent surrogate modelling without labeled data for mesh-free solutions of parametric partial differential equations.
- Time-efficient manufacturing process optimization with surrogate models using physics-informed neural networks (PINNs).
- Over 500 times faster optimization of process and material parameters, demonstrated on the example of a thermochemical composite curing process.
- After training, highly accurate solutions are obtained in negligible time with a maximum deviation of 0.16%, compared to 2.4 hours simulation time of the reference finite element solver.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 20 December 2022
Revised 15 May 2023
Accepted 22 May 2023
Available online 26 May 2023

Keywords:
Machine learning
Physics-based modeling
Mesh-free surrogate modeling
Data-free surrogates
Composite process optimization

ABSTRACT

Engineering components require an optimization of design and manufacturing parameters to achieve maximum performance – usually involving numerous physics-based simulations. Optimizing these parameters is a resource-intensive endeavor, though, especially in high-dimensional scenarios or for complex materials like fiber reinforced plastics. Surrogate models are able to reduce the computational effort, however, data generation still proves to be resource-intensive. Additionally, their data-driven nature may lead to physically implausible results in limit cases. As a remedy, physics-informed neural networks (PINNs) include known physics into the training for enhanced surrogate reliability. This allows to cast a physically consistent, data- and mesh-free manufacturing surrogate for variable process conditions and material parameters. The paper demonstrates how PINNs can be embedded in a design-framework to enhance process understanding, to devise engineering-interpretable processing windows and to support time-efficient process optimization at the example of a thermochemical manufacturing process with fiber-reinforced composite materials. In this work, an over 500-fold speed up of the process optimization is achieved compared to conventional approaches.

* Corresponding author.
E-mail address: tobias.wuerth@kit.edu (T. Würth).

https://doi.org/10.1016/j.matdes.2023.112034
0264-1275/

1. Introduction and related work

Individualized products, shrinking lot sizes and shorter development times demand a wide variety of rapidly-developed products. However, even small product adaptions imply a new optimization of the underlying manufacturing process to ensure a good part quality and cost-efficiency. In current practice, such optimizations entail costly trial–error experiments, especially for a complex manufacturing process, such as processing fiber reinforced plastics materials.

An optimization of product properties can be considered as the search of manufacturing process and design parameters, which maximize defined product goals, while satisfying product and manufacturing constraints [1]. In order to reduce experimental efforts, high-fidelity simulations are applied, e.g. finite element (FE) models. They allow concentrating costly experiments on the most promising variants, however, finding a process optimum by iterative optimization can still be time-consuming, due to the required large number of evaluations in the usually non-linear and non-convex optimization landscape [2,3].

Possible relief might come through the use of surrogate-based optimization (SBO) [3]. Surrogates are numerically efficient, data-driven models, which guide the optimizer's search of the parameter space. Frequently used surrogate models are e.g. polynomial response surface (PRS), radial basis function (RBF), support vector regression (SVR), Gaussian process and neural network models [4]. On the downside, they usually require large data sets and can be prone to physically implausible results in limit cases.

First introduced in 2017 by Raissi et al. [5], PINNs recently got successfully applied as a surrogate model to different engineering tasks, such as linear elasticity and nonlinear elastoplasticity [6], fluid dynamics [7] and a curing process of composite materials [8]. Unlike conventional surrogate models, PINNs, as a physics-informed deep learning method, can act like a numerical solver and solve forward problems without labeled data [8–10]. Thereby, the data-free surrogate model can be obtained, by solving the considered partial differential equations (PDEs) directly parameterized, e.g. as a function of variable material and process parameters [11,7]. Consequently, PINNs have the potential to combine the physical consistency of classical numerical simulations with the time-efficiency of surrogates.

Additionally, PINNs provide further advantages. If data is available, such as numerical or experimental data, the solving of the PDEs can be accelerated [12]. In case of small model adaptions, previous PINNs can be re-used (transfer learning) to reduce training time and computational efforts [13]. Due to the regarded physic in the training, a well-trained PINN can produce accurate results with little data [6], unlike purely data-driven machine learning models, which often lack robustness in sparse-data situations [14]. Furthermore, a mesh-free surrogate model implementation can be evaluated in a fraction of a second (on Intel i5 9500©at 3.00 GHz) and the challenging and problem-dependent effort of a mesh-generation is circumvented [7]. The efficiency of PINNs even multiplies during repetitive evaluations, e.g. during optimization or uncertainty analysis [13]. The savings can be several orders of magnitude compared to conventional numerical solvers. In cases of changed optimization goals, e.g. changed material parameters, a PINN surrogate model can be used again without the need of another training. Overall, PINNs appear a promising option for engineering design and manufacturing optimization.

In this work, a manufacturing optimization approach is proposed that considers manufacturing process and material properties, by using PINNs as a numerical solver and data-free surrogate model. Unlike prior work, this work not just applies PINNs to solve physics-based PDEs but embeds them in a holistic engineering framework for improved process understanding and efficient optimization. In addition, the performance gain compared to classical simulation-assisted optimization is quantified. The framework is demonstrated on a two-way coupled non-linear PDE of application relevant complexity. Additionally, the potential of PINNs for engineering application is further outlined by simultaneously varying initial conditions, boundary conditions (process parameters) and field equation (material parameters). The optimization process is divided into two main steps: Firstly, a surrogate model is obtained by solving the parameterized PDEs with PINNs. Secondly, the trained PINNs are used to optimize the manufacturing process by variation of process and material parameters. Furthermore, it is demonstrated how the influence of process and material parameters can be conveniently analyzed within this framework, which helps to receive physically reasonable insights into the modeled manufacturing process. This allows the engineer to classify and assess the quantitative optimization results.

The surrogate modeling method is firstly evaluated on a generic 1D heat transfer problem to develop an understanding of the method's capability with parameterized PDEs. Then, the optimization method is applied on a simplified manufacturing process: a thermochemical curing process of a composite plate. Subsequently, the trained surrogate model is used to minimize the manufacturing costs of the plate.

The paper is organized as follows: In Section 2 the overall methodology of the proposed optimization method is outlined. Section 3 at first illustrates parametric PDEs and their solution with PINNs on a generic example before modeling and optimizing a composite manufacturing process. In Section 4 the results of the work are summarized, discussed and concluded by an outlook to future developments.

2. Solving parameterized PDEs with PINNS

2.1. Approximation of parameterized PDEs by surrogate models

In general, a manufactured product can be characterized by a tuple of product properties $\underline{p}(\underline{\xi})$, which depend on a tuple of processing and design parameters $\underline{\xi} \in \underline{\Xi}$. Modeling the product property function $\underline{p}(\underline{\xi})$, described by a system of PDEs, is desirable for several engineering tasks, like design optimization or uncertainty quantification [13], or to gain knowledge about the production process itself. A final product property $\underline{p}(\underline{\xi})$ usually depends on the initial state and the evolution of state variables $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$, like plastic strains or residual stresses. Here, $\boldsymbol{x}$ denotes the spatial vector and t the time variable. In this work, underlined quantities $\underline{q}$ denote array-like structures, while quantities in bold $\boldsymbol{q}$ are tensors. In general, it is necessary to solve a system of coupled parameterized PDEs, in order to obtain a numerical solution of the state variables $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$ and, as a consequence, of the product properties $\underline{p}(\underline{\xi})$. For an initial-boundary value problem, the system of PDEs contains initial conditions (ICs) $\underline{\mathscr{I}_u}$, Dirichlet and Neumann boundary conditions (BCs) $\underline{\mathscr{B}_u}$ and generally nonlinear PDEs $\underline{\mathscr{D}_u}$:

$$\begin{cases} \underline{\mathscr{D}_u}(\underline{u}, \boldsymbol{x}, t, \underline{\xi}) = \boldsymbol{0}, & \boldsymbol{x} \in \Omega, t \in \mathscr{T}, \underline{\xi} \in \underline{\Xi} \\ \underline{\mathscr{B}_u}(\underline{u}, \boldsymbol{x}, t, \underline{\xi}) = \boldsymbol{0}, & \boldsymbol{x} \in \partial\Omega, t \in \mathscr{T}, \underline{\xi} \in \underline{\Xi} \\ \underline{\mathscr{I}_u}(\underline{u}, \boldsymbol{x}, t = t_0, \underline{\xi}) = \boldsymbol{0}, & \boldsymbol{x} \in \Omega, \underline{\xi} \in \underline{\Xi} \end{cases} \tag{1}$$

where Ω denotes the spatial problem domain, $\partial\Omega$ the boundary of the spatial domain, $\mathscr{T} = [t_0, t_1]$ the time space and $\underline{\Xi}$ a tuple of parameter spaces of the parameters $\underline{\xi}$.

Due to the complexity, an analytical solution of the function $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$ is usually inaccessible. Conventional numerical techniques

can be used to approximate discrete solutions of the system of PDEs, but require substantial computational effort. A potential remedy are data-driven surrogate models, which yield a continuous approximation of the searched function $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$. However, being entirely statistical models, such function approximations can be physically inconsistent, though. In contrast, PINNs can create a physics-based surrogate model of the function in a single step, by solving the PDEs parameterized, without the need of numerical solvers [7].

2.2. Physics-informed neural networks

Traditional numerical solvers provide a solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi}^*)$ for one distinct parameter combination $\underline{\xi}^*$ only. In contrast, PINNs can solve the PDEs directly, providing a parameterized solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$. An example architecture of such a PINN, using a fully-connected feed-forward neural network, is depicted by Fig. 1. Like proposed in [6], an individual neural network is used for each searched quantity $u_i(\boldsymbol{x}, t, \underline{\xi})$ of the tuple $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$. It receives the arguments $\boldsymbol{x}, t$ and $\underline{\xi}$ as input and yields a scalar output $\tilde{u}_i(\boldsymbol{x}, t, \underline{\xi}; \underline{\theta}_i)$, depending on the weights and biases $\underline{\theta}_i$ of the network. The tangens hyperbolicus function tanh is used as the activation function for all layers, except for the output layer, which activation function is selected as one. Corresponding to the universal approximation theorem, neural networks can approximate the parameterized solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$ for any desired accuracy, if the network size is not restricted [15]. In order to model the solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi}) = \underline{\tilde{u}}(\boldsymbol{x}, t, \underline{\xi}; \underline{\theta}^*)$ with neural networks, the appropriate weights and biases $\underline{\theta}^*$ of all networks have to be found (network training).

A PINN can be trained with or without labeled data. In addition to the data loss $\mathscr{L}_{\text{data}}$, the total loss $\mathscr{L}$ of a soft-constraints PINN approach contains a loss on the PDEs $\mathscr{L}_{\underline{\mathscr{D}}}$, a loss on the BCs $\mathscr{L}_{\underline{\mathscr{B}}}$ and a loss on the ICs $\mathscr{L}_{\underline{\mathscr{I}}}$. As a consequence of the data-free training (without labeled data) in this work, the data loss $\mathscr{L}_{\text{data}}$ of the total loss $\mathscr{L}$ equals automatically to zero. For $N_{\underline{\mathscr{D}}}$ training samples on the domain, $N_{\underline{\mathscr{B}}}$ training samples of the BCs and $N_{\underline{\mathscr{I}}}$ training samples of the ICs, the total loss can be calculated with the following equations:

$$\begin{cases} \mathscr{L} = \lambda_{\underline{\mathscr{D}}}\mathscr{L}_{\underline{\mathscr{D}}} + \lambda_{\underline{\mathscr{B}}}\mathscr{L}_{\underline{\mathscr{B}}} + \lambda_{\underline{\mathscr{I}}}\mathscr{L}_{\underline{\mathscr{I}}} \\ \mathscr{L}_{\mathscr{S}} = \frac{1}{N_{\mathscr{S}}}\sum_{i}^{N_{\mathscr{S}}} |\mathscr{S}_{\underline{u}}(\tilde{\mathbf{u}}, \boldsymbol{x}_i, t_i, \underline{\xi}_i)|^2, \quad \mathscr{S} = \underline{\mathscr{D}}, \underline{\mathscr{B}}, \underline{\mathscr{I}} \end{cases} \tag{2}$$

wherein the weights $\lambda_{\mathscr{S}}$ define the contribution of the individual losses to the total loss and are set to one in this work. For each training sample $(\boldsymbol{x}_i, t_i, \underline{\xi}_i)$ the output of the neural network $\tilde{u}(\boldsymbol{x}_i, t_i, \underline{\xi}_i)$ and its derivatives with respect to space $\boldsymbol{x}$ and time t have to be calculated and inserted into equations $\mathscr{S}_{\underline{u}}(\tilde{\mathbf{u}}, \boldsymbol{x}_i, t_i, \underline{\xi}_i)$ of the initial-boundary problem, e.g. into the PDE $\mathscr{D}_{\underline{u}}(\tilde{\mathbf{u}}, \boldsymbol{x}_i, t_i, \underline{\xi}_i)$ for a training sample located within the domain. The partial derivatives of the solution $\tilde{u}(\boldsymbol{x}_i, t_i, \underline{\xi}_i)$ can be calculated explicitly and numerically exactly with automatic differentiation (AD) [7] for each training sample. For a non-trained PINN, the tested equations $\mathscr{S}_{\underline{u}}(\tilde{\mathbf{u}}, \boldsymbol{x}_i, t_i, \underline{\xi}_i)$ differ generally from the (analytically) exact solution zero, resulting in a non-zero total loss $\mathscr{L}$. Thus, the solution of the system of PDEs $\underline{u}(\boldsymbol{x}, t, \underline{\xi}) = \underline{\tilde{u}}(\boldsymbol{x}, t, \underline{\xi}; \underline{\theta}^*)$ can be found, by finding the weights and biases $\underline{\theta}^*$ of the neural networks, which minimize the total loss $\mathscr{L}$:

$$\underline{\theta}^* = \arg\min_{\underline{\theta} \in \underline{\Theta}} \mathscr{L}(\underline{\theta}) \tag{3}$$

wherein $\underline{\Theta}$ denotes the spaces spanned by the weights and biases $\underline{\theta}$ of the neural networks. In Appendix A.1 the implementation of the PINNs and their training are described in detail.

2.3. Solution analysis

After the training, a parameterized solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$ of the considered problem is obtained. In most cases, the aim of solving a parameterized system of PDEs is not the solution itself, but rather the use of the solution for further analysis, e.g. a design optimization. In such an iterative solution scenario, the architecture of the PINNs offers additional advantages: Since the PINN-solution is parameterized and continuous for all input variables, it can be evaluated at any desired space $\boldsymbol{x}$ and time t for any parameter combination $\underline{\xi}$ in a fraction of a second. Likewise, derivative-dependent quantities can be calculated exactly and efficiently by automatic differentiation [7].

Overall, this provides a scientific environment, where problems, such as a manufacturing process, can be calculated quickly, accurately and parameterized. This allows the inspection not just of the solution variable u but also of change rates, gradients, Jacobian matrices, Hessian matrices or physical variables like a flux, a momentum and any others.

2.4. Process and design optimization

The goal of a design optimization is to adapt the product characteristics $\underline{p}(\underline{\xi})$ in such a way that a particular optimization objective $\mathscr{G}(\underline{p}(\underline{\xi}))$ is achieved. Consequently the optimization process can be formulated as the search of a tuple of parameters $\underline{\xi}_{\text{min}}$, which minimizes the optimization objective:

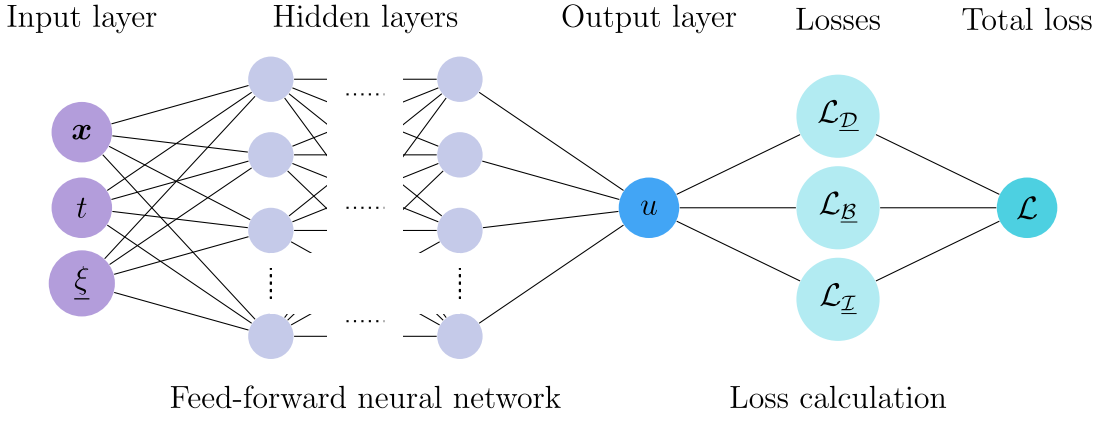


Fig. 1. Example architecture of a physics-informed neural network.

$$\underline{\xi}_{\min} = \arg\min_{\underline{\xi}\in\underline{\Xi}} \mathcal{G}(\underline{p}(\underline{\xi})) \tag{4}$$

The parameterized PINN-solution $\underline{u}(\boldsymbol{x}, t, \underline{\xi})$ allows to compute the product characteristics $\underline{p}(\underline{\xi})$ and the combination with optimization algorithms, e.g. of the scipy.optimize package [16]. Owing to the negligible evaluation time of the PINN, even iteration-intensive optimizers converge to a solution in short time. Some optimizers additionally benefit from the Jacobian or Hessian matrix of the product characteristics, which is readily available through automatic differentiation. Overall, the duration of the optimization process can be neglected in relation to the effort of the training process. Thus, even frequent changes of the optimization objective, e.g. due to product adaptations, can be handled efficiently.

3. PINNs for composite process analysis and design

Section 3 investigates two example applications of PINNs to obtain parametric solutions and explores their advantages. At first, a generic 1D heat-transfer scenario is discussed to highlight the difference between the non-parametric and a parametric solution (Section 3.1). Two scenarios are considered: A non-parametric scenario where the PINN solves only one specific equation, which is then extended to a problem with four parameters in the initial and boundary conditions. Then, the method is applied to a composite manufacturing process (Section 3.2) and used for optimization.

3.1. 1D parameterized heat transfer

3.1.1. Problem definition

In a 1D scenario the heat transfer equation is defined as [17]:

$$\frac{\partial T}{\partial t} = a\frac{\partial^2 T}{\partial x^2}, \qquad x \in \mathcal{X}, t \in \mathcal{T}. \tag{5}$$

Here, a denotes the thermal diffusivity, whereby $\mathcal{X} = [x_0, x_1] \subset \mathbb{R}^1$ denotes the domain and $\mathcal{T} = [t_0, t_1]$ the time space.

In general, the initial condition and the Dirichlet boundary conditions of the problem can be described as a function of the tuple of N_ξ parameters $\underline{\xi} = (\xi_1, \xi_2, \ldots, \xi_{N_\xi}) \in \underline{\Xi}$:

$$T(x, t=0, \underline{\xi}) = T_0(x, \underline{\xi}), \quad T(x = x_i, t, \underline{\xi}) = T_{x_i}(x = x_i, t, \underline{\xi}), \quad i = 0, 1 \tag{6}$$

In the following, the problem is considered in $\mathcal{X} = [-1, 1]$m and $\mathcal{T} = [0, 1]$s with the diffusivity defined as $a = 1/\pi \mathrm{m}^2/\mathrm{s}$.

3.1.2. Non-parametric problem

In the first example the initial and boundary conditions are parameter-free, i.e. fixed and invariable:

$$\begin{aligned} T(x, t=0) &= (T_{x_0} - T_{x_1}) \cos\left(\tfrac{\pi}{2}\tfrac{x-x_0}{x_1-x_0}\right) + T_{x_1}, \\ T(x = x_0, t) &= T_{x_0}, \quad T(x = x_1, t) = T_{x_1}, \end{aligned} \tag{7}$$

whereby $T_{x_0} = 20°\mathrm{C}$ and $T_{x_1} = 0°\mathrm{C}$.

The settings of the PINN network and training are listed in Table 1. The obtained solution $T(x, t)$ is compared with a numerical solution $T_{\mathrm{FDM}}(x, t)$, calculated using the finite difference method (FDM) and evaluated for the 1D heat equation. The relative error of the PINN solution is quantified by:

$$\epsilon_{\mathrm{rel}} = \frac{T(x, t) - T_{\mathrm{FDM}}(x, t)}{T_{x_0} - T_{x_1}} \tag{8}$$

Figs. 2 a) and c) show the PINN solution $T(x, t)$ and the FDM solution $T_{\mathrm{FDM}}(x, t)$, respectively, as a function of x and t. In Fig. 2 b) those solutions are compared for different time steps $t_i = (0.0\mathrm{s}, 0.3\mathrm{s}, 1.0\mathrm{s})$ and the time evolution of the solution is shown: At the beginning, the temperature distribution in the domain has the shape of a cosine function from 0 to $\pi/2$. Due to thermal diffusion, the temperature distribution gradually approaches the steady-state solution: a linear decrease from T_{x_0} to T_{x_1}. For all three time steps, the PINN-solution matches the FDM-solution well ($\max(|\epsilon_{\mathrm{rel}}|) < 0.4\%$) as Fig. 2 d) shows. The maximum and minimum relative error of the PINN solution is located on the initial and left boundary condition.

3.1.3. Four-parameter problem

In this section, the problem is solved for four parameters $\underline{\xi} = (\xi_0, \xi_1, \xi_2, \xi_3)$ to see how the PINNs can handle multiple parameters. The parameters $\underline{\xi}$ enable a wide variety of initial and corresponding boundary conditions:

$$\begin{aligned} T(x, t=0, \underline{\xi}) &= \xi_2(x - x_0)(x - x_1)(x - \xi_3) + \xi_0 + (\xi_1 - \xi_0)\tfrac{x-x_0}{x_1-x_0}, \\ T(x = x_0, t, \underline{\xi}) &= \xi_0, \quad T(x = x_1, t, \underline{\xi}) = \xi_1. \end{aligned} \tag{9}$$

Thereby, the parameters ξ_0 and ξ_1 are considered in the space $\Xi_0 = [0, 20]°\mathrm{C}$, ξ_2 in $\Xi_2 = [-10, 10]°\mathrm{C/m}^3$ and ξ_3 in $\Xi_3 = [-2, 2]\mathrm{m}$.

The network and training settings of the PINN are listed in Table 1. Once again, the results are compared to the FDM solution described above.

The comparison of the solutions for three randomly defined parameter combinations $\underline{\xi}_i = (\xi_{0,i}, \xi_{1,i}, \xi_{2,i}, \xi_{3,i})$ is shown at the time steps $t = 0.0\mathrm{s}, t = 0.3\mathrm{s}$ and $t = 1.0\mathrm{s}$ in Fig. 3 a) to c). For each time step and each parameter combination, the PINN results show a good graphical agreement with the FDM results. Please note that PINNs enable an unlimited amount of solutions evaluations within the definition range of the parameters $\underline{\xi}$. All results shown were produced by a single PINN after a single training, while the FDM method required a new computation for each graph.

3.2. Thermochemical curing process

Previously, the PINN surrogate models were implemented successfully, producing accurate results in the generic example of Section 3.1 with up to 4 parameters. In this section, the method is transferred to a more practical problem of a thermochemical curing process.

3.2.1. Problem definition

In the following, a thermochemical non-isothermal curing process of a carbon fiber reinforced plastic (CFRP) plate is considered, simulating an autoclave process. The treated plate is 250mm wide (x-direction), 4mm high (y-direction) and 800mm long (z-direction), with a carbon fiber fleece as a reinforcement. Isotropic and homogeneous material properties, no temperature changes in z-direction and a constant processing pressure are assumed. For a 2D isotropic heat transfer of a composite curing process, the governing equation can be defined as [18]

$$\frac{\partial}{\partial t}(\rho_{\mathrm{c}} c_{\mathrm{p,c}} T) = k_{\mathrm{c}}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + (1 - c)\rho_{\mathrm{m}}\Delta h_{\mathrm{m}}\dot{\alpha}. \tag{10}$$

The macroscopic composite properties s, specifically the density ρ_{c}, the specific heat capacity $c_{\mathrm{p,c}}$ at constant pressure and the thermal conductivity k_{c}, depend on the fiber volume content (FVC) c, corresponding to the rule of mixture

$$s_{\mathrm{c}} = c\, s_{\mathrm{f}} + (1 - c) s_{\mathrm{m}}, \quad s = \rho, c_{\mathrm{p}}, k. \tag{11}$$

Constants with the index f are material properties of the carbon fibers, whereas the index m denotes a matrix property. The last term of the Eq. 10 describes the internal heat generation as a result of the exothermal curing kinetics $\dot{S}_T = (1 - c)\rho_{\mathrm{m}}\Delta h_{\mathrm{m}}\dot{\alpha}$. For the curing kinetics, the Kamal-Malkin kinetic model is used [19]:

Table 1
Neural network (top) and training (bottom) settings of the 1D heat transfer and 2D thermochemical problem.

	1D heat transfer		*2D thermochemical problem*	
Network setting	**Section 3.1.2**	**Section 3.1.3**	**Section 3.2.2**	**Section 3.2.3**
Network inputs	x, t	$x, t, \underline{\xi}$	x, y, t	$x, y, t, \underline{\xi}$
Network outputs	T		$T; \alpha$	
Hidden layers	2	4	4	8
Neurons per hidden layer	20		20	40
Training setting	**Section 3.1.2**	**Section 3.1.3**	**Section 3.2.2**	**Section 3.2.3**
Number training samples	10^5	10^7	10^6 (T); 2×10^6 (α)	
Distribution samples	50/25/12.5 % (D/IC/BC-LR)		50/25/6.25 % (D/IC/BC-LRTB)	
Training epochs	500		500	
Batch size	1000		1000	
Learning rate	10^{-3} to 10^{-5} (ED)		10^{-3} to 10^{-5} (ED)	

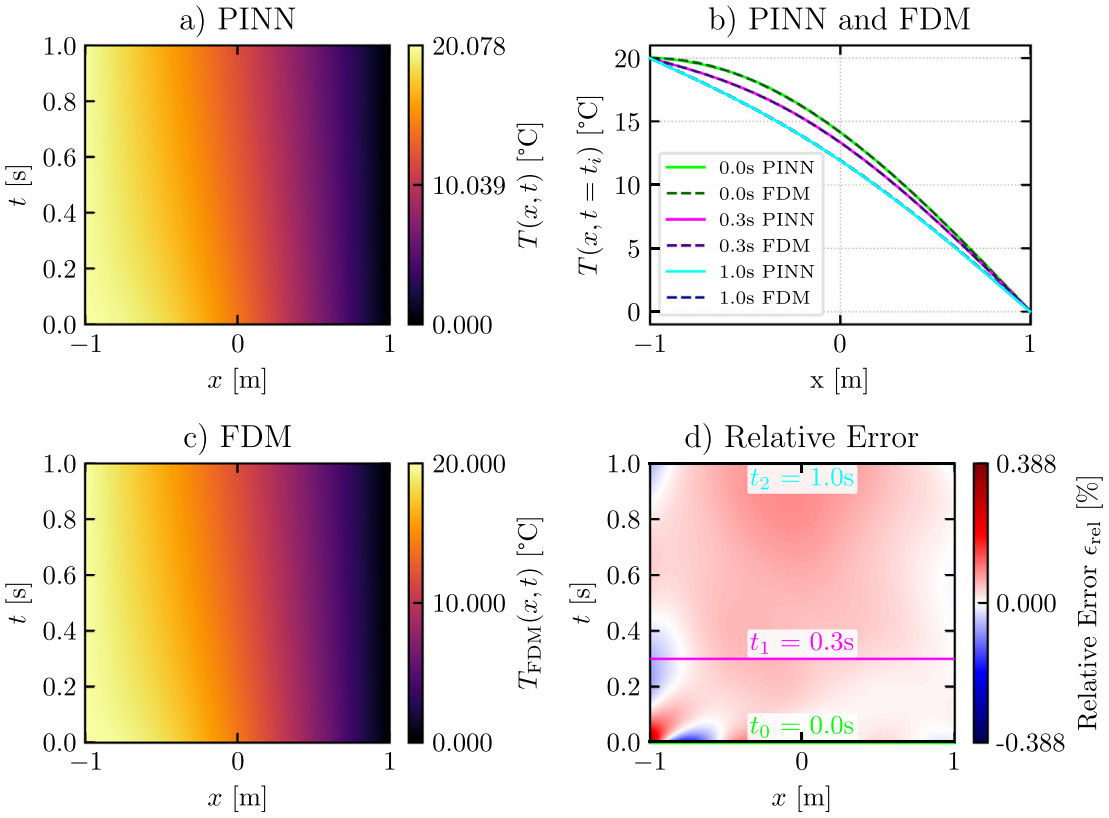


Fig. 2. Solution of the PINN for the 1D non-parametric initial and boundary conditions of Section 3.1.2 compared to a reference solution obtained by using the finite difference method. Figure a) shows the PINN solution of the temperature $T(x, t)$. In Figure b) is the PINN solution (continuous lines) of the temperature $T(x, t = t_i)$ is compared to the corresponding reference solution (dashed lines), both evaluated at different time steps $t_i = (0.0\text{s}, 0.3\text{s}, 1.0\text{s})$ (green, pink, respectively blue lines). Figure c) shows the reference solution of the temperature $T_{\text{FDM}}(x, t)$ obtained by the FDM. In Figure d) the relative error ϵ_{rel} of the PINN solution in relation to the FDM solution is plotted.

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = (K_1 + K_2\alpha^m)(1-\alpha)^n. \tag{12}$$

Thereby, the reaction rate constants K_1 and K_2 follow the Arrhenius identity $K_i = A_i \exp(-E_i/(RT))$, wherein A_i denotes a speed of reaction constant, E_i an activation energy and R the universal gas constant [19,20].

At the beginning of the process, the temperature in the whole plate is constant, $T(x, y, t = 0, \underline{\xi}) = T_0 = 20°\text{C}$, and the curing has not started, $\alpha(x, y, t = 0) = \alpha_0 = 0$. In the considered autoclave process (see Fig. 4), the process temperature is increased over time up to a holding temperature T_1. A resulting time-dependent temperature at the top of the plate is assumed, described by the following equation:

$$T(x, y = y_1, t, \underline{\xi}) = (T_1 - T_0)\tanh\left(\frac{t}{\tau_0}\right) + T_0. \tag{13}$$

The constant $\tau_0 = t_{\text{heat}}/(\text{artanh}(s_0))$ is chosen so that the tanh takes the value $s_0 = 0.99$ at $t = t_{\text{heat}} = 20\text{s}$

Two channels cool the plate from underneath with a constant heat flux of:

$$\begin{aligned} Q_y(x, y = y_0, t, \underline{\xi}) &= Q_0\left(\sin\left(\tfrac{2\pi(x-x_0)}{x_1-x_0} - \tfrac{\pi}{2}\right) - s_1\right)\tanh\left(\tfrac{t}{\tau_0}\right), \\ Q_y(x, y = y_0, t, \underline{\xi}) &= -k_c \frac{\partial T(x, y, t, \underline{\xi})}{\partial y}\Big|_{y=y_0}, \end{aligned} \tag{14}$$

with the constants $Q_0 = 5000\,\frac{\text{W}}{\text{m}^2}$ and $s_1 = 3$.

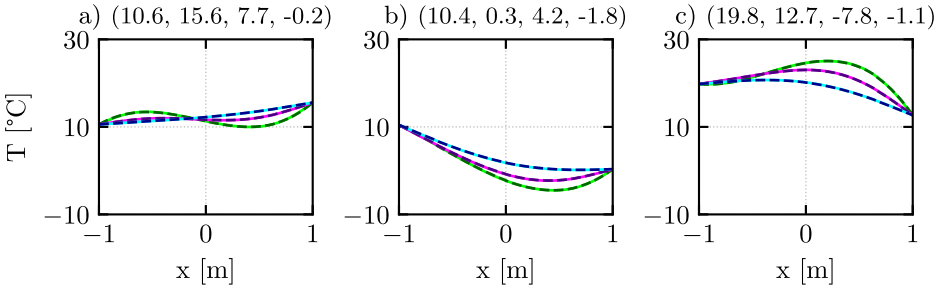


Fig. 3. The PINN $T(x,t,\underline{\xi})$ and FDM $T_{\text{FDM}}(x,t,\underline{\xi})$ solutions for the 1D heat transfer problem described in Section 3.1.3, which contains 4 parameters in total. Every Figure from a) to c) shows the solution of the PINN (continuous lines) and FDM (dashed lines) for a distinct parameter combination $\underline{\xi}_i = (\xi_{0,i}, \xi_{1,i}, \xi_{2,i}, \xi_{3,i})$ at the time steps $t = 0.0$s (green lines), $t = 0.3$s (pink lines) and $t = 1.0$s (blue lines). Each parameter combination $\underline{\xi}_i$ is randomly defined. In the title of the subplots the corresponding parameter combination can be seen in the format $(\xi_{0,i}, \xi_{1,i}, \xi_{2,i}, \xi_{3,i})[^\circ\text{C}, ^\circ\text{C}, ^\circ\text{C}/m^3, \text{m}]$.

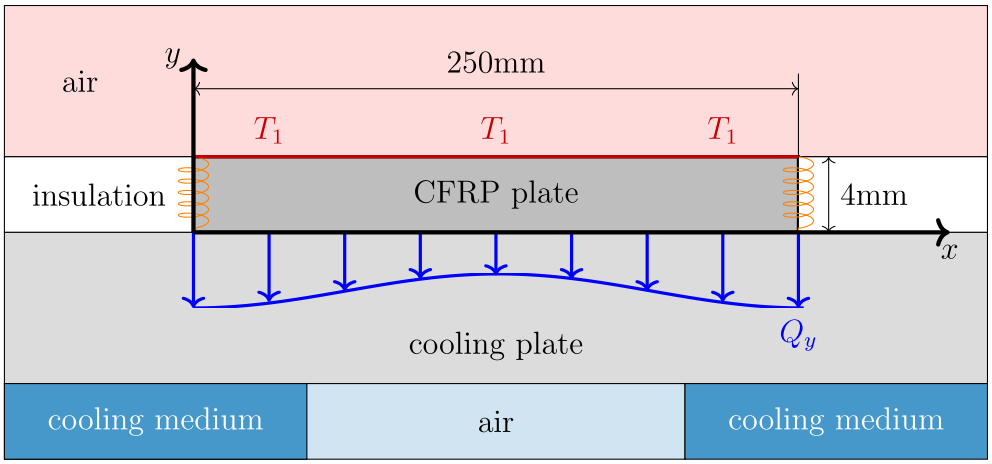


Fig. 4. Thermochemical curing process of a CFRP plate, considered in Section 3.2.

Both the left and the right side are adiabatic, which can mathematically be formulated as:

$$\left.\frac{\partial T(x,y,t,\underline{\xi})}{\partial x}\right|_{x=x_0} = \left.\frac{\partial T(x,y,t,\underline{\xi})}{\partial x}\right|_{x=x_1} = 0. \tag{15}$$

The problem equation is considered for $\mathbf{x} = (x,y) \in \mathscr{X} = [x_0, x_1] \times [y_0, y_1] = [0, 250]\text{mm} \times [0, 4]\text{mm}$ and $t \in \mathscr{T} = [t_0, t_1] = [0, 100]\text{s}$.

3.2.2. Non-parametric problem

In the first step, the thermochemical curing problem is solved without a parameter dependence. The maximum heating temperature on the top is selected as $T_1 = 120$ °C and an FVC of $c = 50\%$. With the value of c, the material properties of the composite material can be calculated using the Eqs. (11). The material properties are listed in Table 2. Table 1 shows the network and training settings of the PINNs.

The PINN-solution needs to be validated. For lack of available analytical solution or spatially and temporally sufficiently resolved experiments, the obtained PINN-solution is compared with a reference FEM solution. Specifically, the built-in thermal solver of the commercial FE-package Abaqus is used to solve the heat equation as it is a well-known and widely accepted solver. For the material model of the curing kinetics, a validated subroutine is used. Details on the material model and according experimental validation are given in the references [21]. The relative error for the temperature and the curing degree is defined as:

$$\begin{aligned} \epsilon_{\text{rel},T} &= \frac{T(x,y,t) - T_{\text{FEM}}(x,y,t)}{T_1 - T_0}, \\ \epsilon_{\text{rel},\alpha} &= \alpha(x,y,t) - \alpha_{\text{FEM}}(x,y,t). \end{aligned} \tag{16}$$

Table 2
Materials properties of the thermochemical curing process [23–28,21] in section 3.2

Property	Symbol	Value	Unit
Thermal properties			
Thermal conductivity (carbon fiber)	k_{f}	2.40e1	W/(m K)
Density (carbon fiber)	ρ_{f}	2.25e3	kg/m^3
Specific isobaric heat capacity (carbon fiber)	$c_{\text{p,f}}$	7.08e2	J/(kg K)
Thermal conductivity (matrix)	k_{m}	2.90e-1	W/(m K)
Density (matrix)	ρ_{m}	1.24e3	kg/m^3
Specific isobaric heat capacity (matrix)	$c_{\text{p,m}}$	2.00e3	J/(kg K)
Properties of curing reactions			
Speed of reaction constant	A_1	5.38e10	1/s
Speed of reaction constant	A_2	3.10e3	1/s
Activation energy	E_1	8.63e4	J/mol
Activation energy	E_2	8.00e4	J/mol
Reaction order	m	1.02e-2	–
Reaction order	n	1.15	–
Specific total reaction enthalpy	Δh_{m}	1.89e5	J/kg
Gas constant	R	8.314	J/(K mol)

Figs. 5 (a) and (b) depict the PINN's solution of the temperature field T and the curing degree, respectively, in the plate at the time step $t = 20$s. At this time step, the heated top of the plate has reached approximately 99% of the heating temperature $T_1 = 120$°C according to the boundary condition. A hot spot can be observed in the center of the plate (cf. P_1 in Fig. 5 a)), which is induced by the heat of the curing reaction. The lower left and right corners are relatively cold, due to the cooling heat flux of the boundary condition in Eq. 14, which has its maximum value

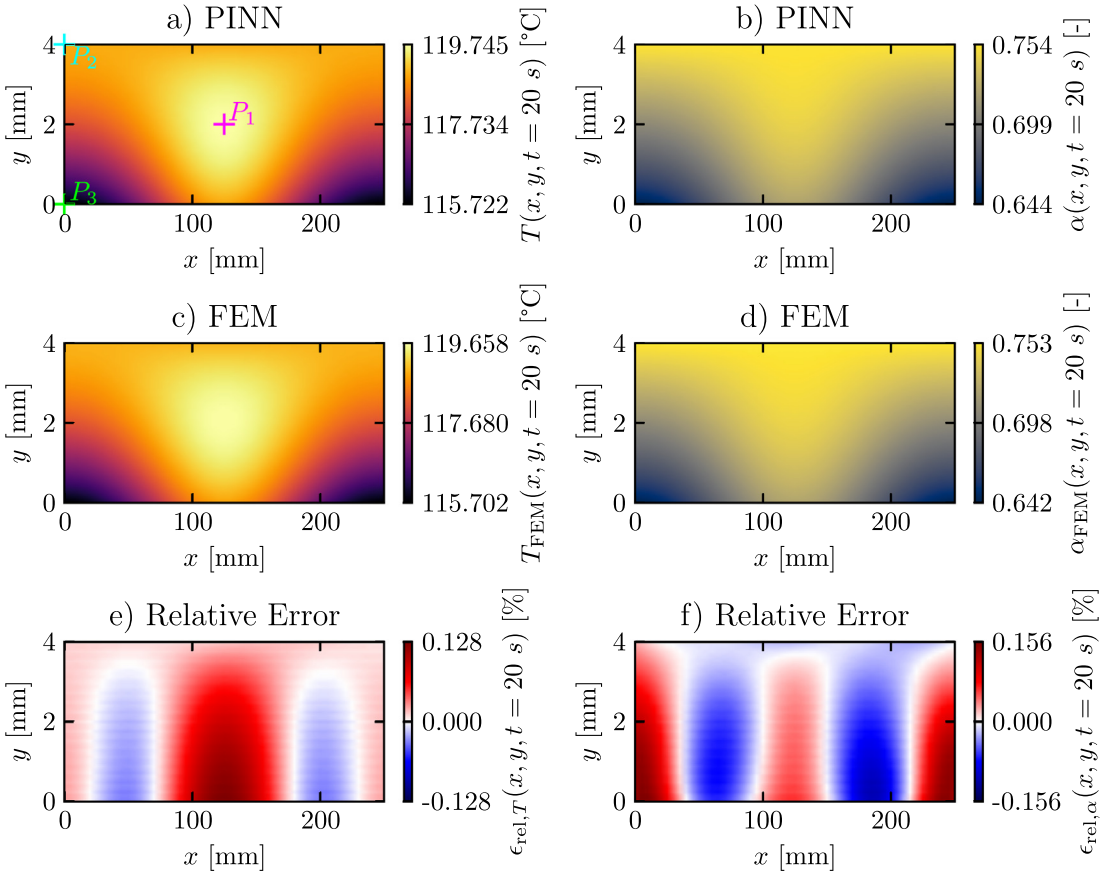


Fig. 5. Solution of the PINNs for the temperature $T(x, y, t)$ and the curing degree $\alpha(x, y, t)$of the problem described in Section 3.2.2 and compared to an FEM solution. Figure a) shows the PINN solution of the temperature distribution in the plate for the time step $t = 20$s. The three points P_1, P_2 and P_3 mark the locations of the temperature and curing degree evaluations in Fig. 6 a) respectively b). In Fig. 5 c) the corresponding FEM solution is plotted, which is the reference for the relative error in e), calculated with Eq. 16. The Figures b), d) and f) show analogously the PINNs solution, FEM solution and relative error of the curing degree in the plate at the time step $t = 20$s.

at the corners. Similar to the temperature distribution, the highest curing degree is at the top and center of the plate, whereas the bottom corners have the lowest values. In Figure e) and f) the relative errors $\epsilon_{\mathrm{rel},T}$ respectively $\epsilon_{\mathrm{rel},\alpha}$ of Eq. (16) are plotted with the reference FEM solution pictured in c) respectively d). The PINNs solve the problem accurately, with an overall maximum relative error of less than 0.2%.

In Fig. 6 the time evolution of the temperature and the curing degree are plotted at the points P_1, P_2 and P_3, which are marked in Fig. 5 a). The PINNs solution shows a good agreement with the FEM solution compared over time. In the temperature curve of P_1 and P_3, hot spots are visible at about 19s, due to an increase in temperature relative to the stationary solution at the end of the process. In Figure b) the curing degree approaches 100% in all three locations, even in the approximately coldest location of the plate P_3.

3.2.3. Two-parameter problem

In this section two parameters are introduced for the problem of Section 3.2.2. The parameter ξ_0 replaces the heating temperature T_1, so that the problem can be evaluated for different processing temperatures in the boundary condition (13) at the top of the plate. The second parameter ξ_1 is a material property, the FVC c, which affects the thermal properties of the composite, e.g. heat capacity and conductivity. As a result, it varies the underlying heat transfer PDE (10) and the released curing reaction heat $\dot{S}_T$.

The problem is investigated for the heating temperature $\xi_0 = [60, 180]$ °C and the FVC $\xi_1 = [25, 75]$ %. As before, the settings of the PINN networks and training is listed in Table 1.

The obtained surrogate models are used to show exemplarily, how PINNs can help analyze the solution dependencies on process and design parameters, to gain deeper insights into the underlying manufacturing and design process. Some influences of the two parameters to the temperature distribution can be recognized in Fig. 7. Subplots g) and h) show a disparate temperature distribution compared to the other plots: They show hot spots; the others not. A hot spot arises, if the curing reaction heat cannot be discharged. This can result in an uncontrollable curing and carbonization. Hot spots can occur at any location the part except for the fixed temperature at the top boundary. For a constant FVC ξ_1 (columns), a higher heating temperature ξ_0 leads, as expected, to higher temperatures. For the same heating temperature ξ_0, lower FVCs ξ_1 result in significantly lower minimum temperatures in the lower part of the plate, if there are no hot spots, cf. Fig. (7) a) to f). Fig. 8 a) illustrates this issue further and shows the relative stationary temperature in the bottom corners at the end of the process $T_{\mathrm{rel,BL,end}}(\xi_0, \xi_1) = (T(x = x_0, y = y_0, t = t_1, \xi_0, \xi_1) - T_0)/(\xi_0 - T_0)$. In loose terms, $T_{\mathrm{rel,BL,end}}(\xi_0, \xi_1)$ quantifies, which proportion of the process temperature rise at the top is reached at the bottom. Thus, one would expect that lower fiber volume contents - and thus lower bottom temperatures - slow down the curing reaction and increase curing time. The time for 99% curing in the bottom corners $t_{99\%}(\xi_0, \xi_1)$ is depicted in Fig. 8 b). While the curing time behaves as expected at lower heating temperatures, the opposite is the case for high heating temperature ξ_0. This contrary effect is confirmed by the higher temperature at the bottom corners in Fig. (7) g). In the following, we show how PINNs can be used to analyze this counter-intuitive effect.

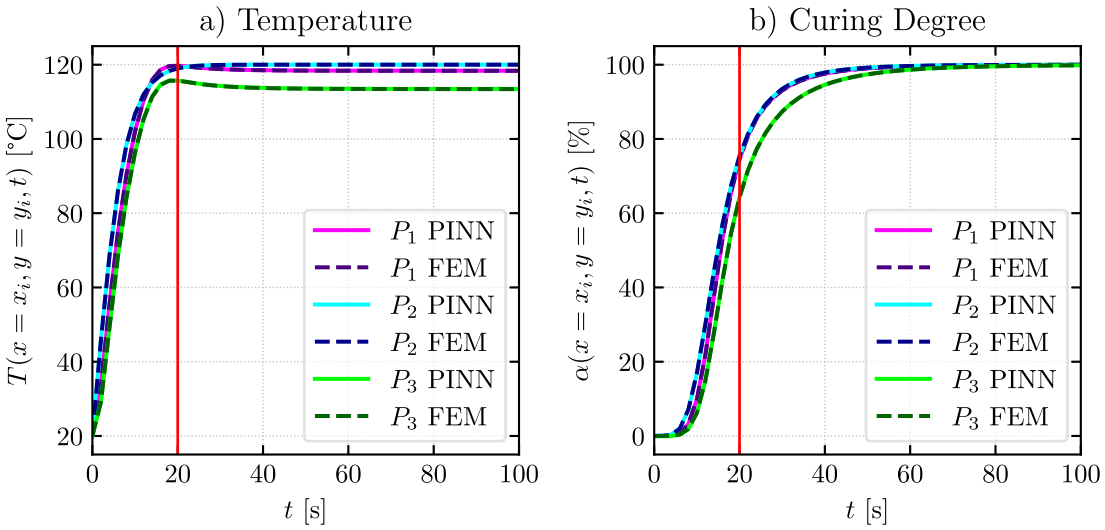


Fig. 6. Comparison over time of PINN and FEM solution for the problem described in Section 3.2.2. Figure a) shows the temperature time evolution and b) the curing degree time evolution. The functions are evaluated at three different locations of the plate, which are marked in Fig. 5 a): P_1 is the midpoint of the plate, P_2 is the top left corner and P_3 is the lower left corner. Because of the symmetry in x-direction, the right corners show the same results as the left side and, consequently, are not plotted.

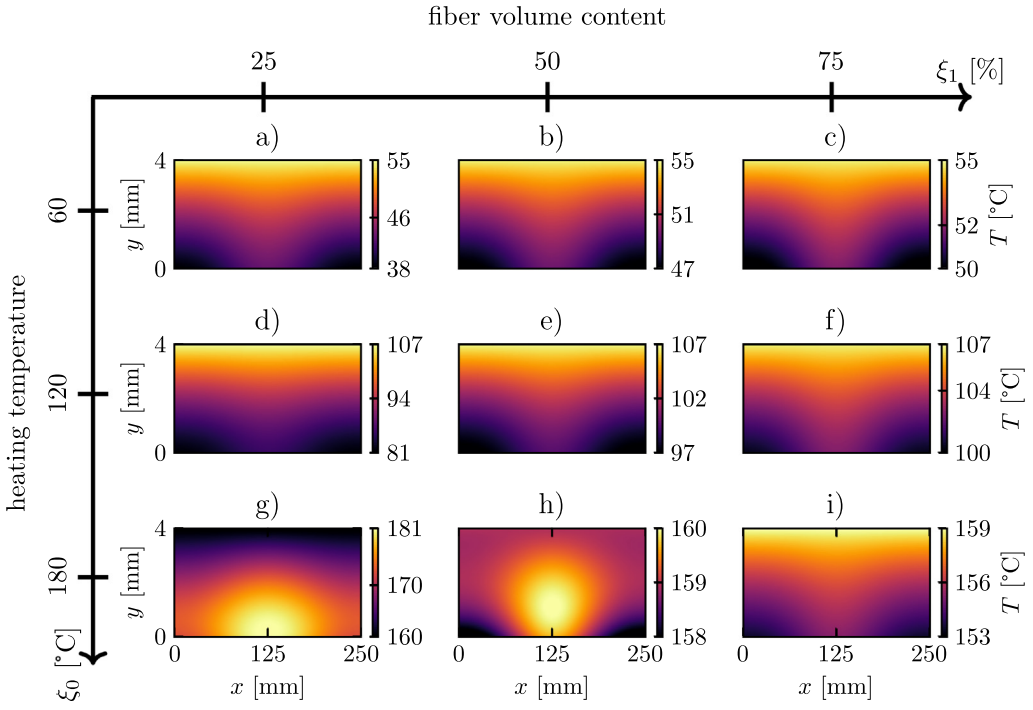


Fig. 7. Temperature distribution in the plate for the problem of Section 3.2.3 approximated by the PINNs and evaluated at the time step $t = 10$s for different parameter combinations. Each Figure from a) to i) shows the solution of temperature distribution for a own parameter combination (ξ_0, ξ_1), which values can be found in the title of the corresponding subplot. The heating temperature ξ_0 increments with each plot from the top to the bottom, while the FVC ξ_1 becomes higher from the left side to the right side.

In order to cure faster, the temperature at the coldest locations (bottom corners) must have been higher at some time during the cycle for low FVCs ξ_1. Indeed, the maximum relative temperature

$$T_{\text{rel,BL,max}}(\xi_0, \xi_1) = \max_{t \in \mathcal{T}} \left(\frac{T(x = x_0, y = y_0, t, \xi_0, \xi_1) - T_0}{\xi_0 - T_0} \right) \tag{17}$$

at the bottom corners during the process (cf. Fig. 8 c)) is higher for low FVCs than for high FVCs. Presumably, arising hot spots are responsible for the discrepancy. In the presence of hot spots, the temperature in the plate is elevated locally. Thus, the maximum relative temperature

$$T_{\text{rel,max}}(\xi_0, \xi_1) = \max_{t \in \mathcal{T}, (x,y) \in \mathcal{X}} \left(\frac{T(x, y, t, \xi_0, \xi_1) - T_0}{\xi_0 - T_0} \right) \tag{18}$$

in the plate during the process shows a raised value for low FVCs ξ_1 and heating temperatures of about 140°C (yellow area in Fig. 8 d)). Without reaction heat, the maximum value would correspond to the heating temperature ξ_0 and, consequently, the maximum relative temperature $T_{\text{rel,max}}(\xi_0, \xi_1)$ would equal to 100% (red area in

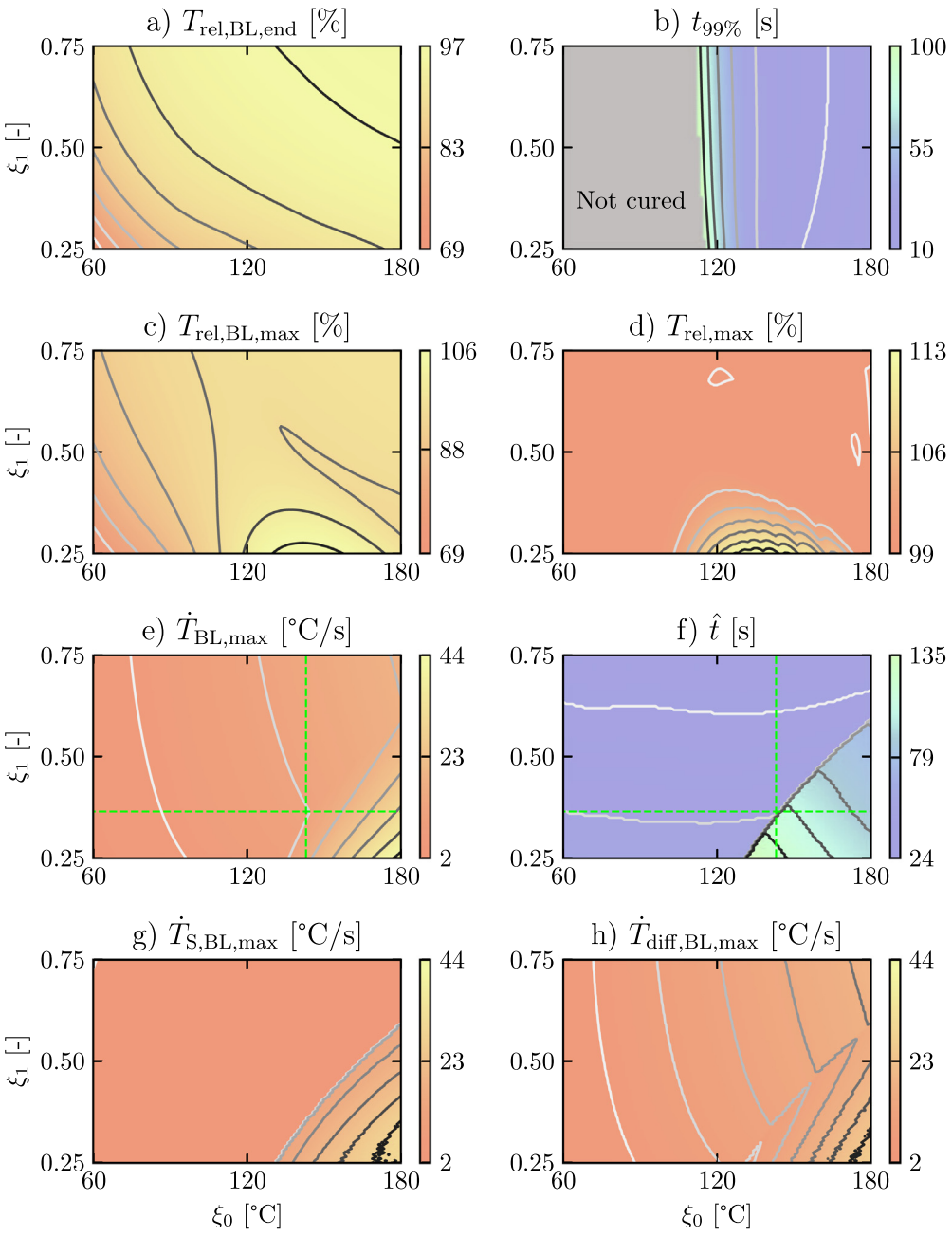


Fig. 8. Evaluation of the PINNs solution. Figure a): relative temperature $T_{\mathrm{rel,BL,end}}$ at the bottom left corner at the end of the process relative to the corresponding heating temperature ξ_0. Figure b): time $t_{99\%}$ for 99% curing in the bottom left corner. Figure c): maximum relative temperature $T_{\mathrm{rel,BL,max}}$ at the bottom left corner. Figure d): maximum relative temperature $T_{\mathrm{rel,max}}$ in the plate. Figure e): maximum rate of temperature change $\dot{T}_{\mathrm{BL,max}}$ in the bottom left corner. Figure f) maximum curing rate $\dot{\alpha}_{\mathrm{BL,max}}$ in the bottom left corner. Figure g) maximum rate of temperature change $\dot{T}_{\mathrm{S,BL,max}}$ in the bottom left corner due to curing heat. Figure h) maximum rate of temperature change $\dot{T}_{\mathrm{diff,BL,max}}$ in the bottom left corner due to diffusion.

Fig. 8 d)). For a deeper analysis of the transient phenomena, the maximum temperature change rate at the bottom corners

$$\dot{T}_{\mathrm{BL,max}}(\xi_0,\xi_1) = \left.\frac{\partial T(x,y,t,\xi_0,\xi_1)}{\partial t}\right|_{x=x_0,y=y_0,t=\hat{t}(\xi_0,\xi_1)} \tag{19}$$

in Fig. 8 e) can be considered, wherein the time function $\hat{t}(\xi_0,\xi_1)$ of the maximum values is defined as

$$\hat{t}(\xi_0,\xi_1) = \underset{t\in\mathcal{T}}{\operatorname{argmax}} \left(\left.\frac{\partial T(x,y,t,\xi_0,\xi_1)}{\partial t}\right|_{x=x_0,y=y_0} \right) \tag{20}$$

and depicted in Fig. 8 f).

For a heating temperature of $\xi_0 = 143$°C (green vertical line), the dependence of the temperature change rate on the FVC switches sharply at an FVC of about 36.5% (green horizontal line). The corresponding time point $\hat{t}(\xi_0 = 143°\mathrm{C}, \xi_1 = 0.356)$ (cf. Fig. 8 f)) is located on an edge, which divides the time function $\hat{t}(\xi_0,\xi_1)$ into an area of small time values (blue-coloured area) at the beginning of the process and an area of higher time values (mint-coloured area). To investigate this behavior, the temperature change rate can be split, in accordance with the heat Eq. (10), into a part related to the curing reaction (cf. Fig. 8 g))

$$\dot{T}_{\mathrm{S,BL,max}}(\xi_0,\xi_1) = \left.\frac{\dot{S}_T(x,y,t,\xi_0,\xi_1)}{\rho_c c_{\mathrm{p,c}}}\right|_{x=x_0,y=y_0,t=\hat{t}(\xi_0,\xi_1)}, \tag{21}$$

and a remaining diffusion change rate (cf. Fig. 8 h))

$$\dot{T}_{\mathrm{diff,BL,max}}(\xi_0,\xi_1) = \dot{T}_{\mathrm{BL,max}}(\xi_0,\xi_1) - \dot{T}_{\mathrm{S,BL,max}}(\xi_0,\xi_1). \tag{22}$$

By comparing the temperature change rate (Fig. 8 e)) with its two parts (Fig. 8 g) and h)) and with the time function $\hat{t}(\xi_0,\xi_1)$ (Fig. 8 f), the parameter dependence can be divided into two areas: For low FVCs and high temperatures, the temperature change rate is dominated by the heat entry of the curing reactions. In contrast, on the other side of the parameter spaces, the dependence in Fig. 8 e) behaves like for the diffusion part of Fig. 8 h). Then, the described sharp switch can be explained by the switch between the areas of the dominant diffusion process respectively the dominant curing reaction heat.

This supports the assumption that the arising hot spots are responsible for the faster curing in Fig. 8 b) for low FVCs ξ_1 and high process temperatures ξ_0: In this process regime, the hot spots form because the heat diffusion is lower than the heat production. Consequently, heat accumulates, raises locally the temperature and initiates the curing reaction. Due to the diffusion process, the temperature in the vicinity of the hot spot increases over time, which in turn initiates curing in neighbouring regions. As a result, the hot spot moves from the center of the plate to the bottom corners.

3.2.4. Process and design optimization

Besides supporting process understanding, PINNs can also be used for efficient process optimization. A simplified cost function of the produced plate is considered. The total production costs of a CFRP plate can be categorized in equipment, tools, labour, energy, consumables and material costs [22]. Some of the costs are fixed costs per plate, like certain labour before and after each plate or consumables of a plate. They do not depend on the curing process itself. Material costs of the fibers and the resin are a function of the FVC ξ_1, whereas the energy costs are coupled with the total energy entry into the plate, due to the heating process on the top, during the process duration t_{pr}:

$$Q_{\mathrm{top}}(t_{\mathrm{pr}},\xi_0,\xi_1) = -z_1 \int_{t_0}^{t_{\mathrm{pr}}} \int_{x_0}^{x_1} k_c \left.\frac{\partial T(x,y,t,\xi_0,\xi_1)}{\partial y}\right|_{y=y_1} \mathrm{d}x\,\mathrm{d}t \tag{23}$$

for a length of the plate $z_1 = 800\mathrm{mm}$. Some equipment and tools costs are dependent on the curing process time t_{pr}, because a higher production volume per equipment can reduce the equipment costs per plate. Because the main purpose of this section is the application of a PINN surrogate model in an optimization scenario, the assumption of a simple cost model is applied:

$$\begin{aligned} C_{\mathrm{plate}}(t_{\mathrm{pr}},\xi_0,\xi_1) = {} & C_t\, t_{\mathrm{pr}} + C_Q\, Q_{\mathrm{top}}(t_{\mathrm{pr}},\xi_0,\xi_1) + C_{\xi_1,\mathrm{f}}\,\xi_1 \\ & + C_{\xi_1,\mathrm{m}}\,(1-\xi_1) + C_0, \end{aligned} \tag{24}$$

including the assumed constants $C_0 = 75€, C_t = 0.5€/\mathrm{s}, C_Q = 200€/(\mathrm{kWh}) = 5.56\times10^{-5}€/J, C_{\xi_1,\mathrm{f}} = 113.64€$ and $C_{\xi_1,\mathrm{m}} = 11.36€$, which values follow the cost distribution of a plate in [22]. The goal of the optimization process is to find the minimum of the cost function:

$$\min_{t_{\mathrm{pr}},\xi_0,\xi_1} C_{\mathrm{plate}}(t_{\mathrm{pr}},\xi_0,\xi_1) \tag{25}$$

with the constraint that the minimum curing degree

$$\begin{aligned} \alpha_{\mathrm{min}}(t_{\mathrm{pr,opt}},\xi_{0,\mathrm{opt}},\xi_{1,\mathrm{opt}}) = {} & \min_{x,y\in\mathcal{X}} \alpha(x,y,t=t_{\mathrm{pr,opt}},\xi_0=\xi_{0,\mathrm{opt}},\xi_1 \\ & = \xi_{1,\mathrm{opt}}) > 99\% \end{aligned} \tag{26}$$

in the part has to be higher than 99% at the end of the process t_{pr}, to guarantee a sufficiently cured plate.

The trust-region algorithm for constrained optimization from [16] is used for optimization. Here, the advantage of neural networks, that the solution output of the network, temperature T and curing degree α, is differentiable with respect to all inputs, can be taken to speed up the optimization by calculating the Jacobian matrix J_C and Hessian matrix H_C of the cost function and of the constraint J_α and H_α w.r.t the variables t_{pr}, ξ_0 and ξ_1. According to the solution of the optimization problem, the lowest FVC $\xi_1 = 25\%$ minimizes the cost function, as a result of the high material costs of the fibers. In the investigated cost model, a lower curing degree is always more cost effective, because it reduces the production time t_{pr} and, in consequence, the heat entry Q_{top}. This happens because mechanical behavior is not considered in the model and thus, the cost function tries to reduce the production cost no matter the structural effects. In Fig. 9 a) the dependence of the curing degree on the two other parameters ξ_0 and t_{pr} is plotted. In Fig. 9 a) the graph show a distinct kink for $90 \lesssim \xi_0 \lesssim 120$ °C. This is due to hot spot formation: their curing reaction heat increases the temperature of the plate. Hence, less energy is required for heat-up. Due to the constraint $\alpha_{\mathrm{min}} = 99\%$, the solving of the optimization problem, dependent on ξ_0 and t_{pr}, can be seen as the search of the minimum of the cost function on the constant curing degree curve $\alpha_{\mathrm{min}} = 99\%$. In Fig. 9 a), where the constant curing degree curves and the optimization solution are plotted into the cost function figure, the optimization solution shows this expected behavior. The solution is approximately located on the

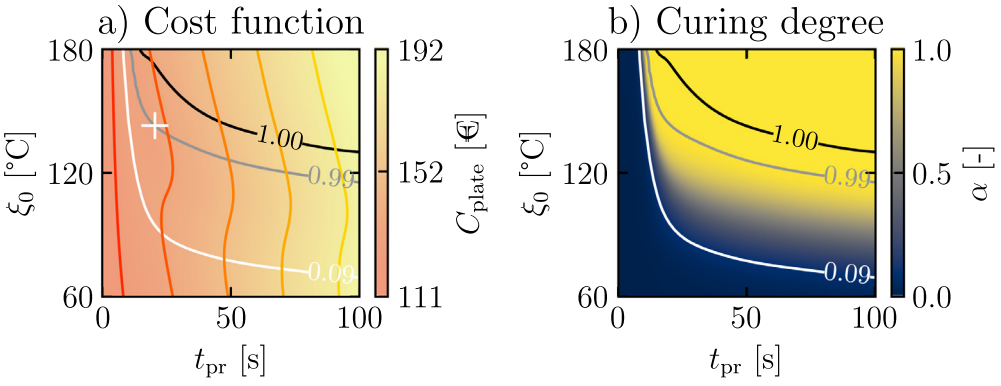


Fig. 9. Optimization solution of the cost function defined in Section 3.2.4 and calculated with PINNs in dependence on the two parameters t_{pr} and ξ_0 for an FVC of $\xi_1 = 25\%$. Figure a) and b) show the cost function and the minimum curing degree α_{min} in the part dependent on the parameters t_{pr} and ξ_0. In Figure a) the constant curing degree curves of Figure b) (black/gray/white lines) and the solution of the optimization (white marker) are plotted into the cost function plot.

curing degree curve $\alpha_{\min} = 99\%$, where the cost function is minimal.

The optimization takes 31.72min on Intel i5 9500©at 3.00 GHz, whereby 120 function, 34 Jacobian and 34 Hessian matrix evaluations of the cost function in total are necessary. The mean simulation time of an FEM simulation is 2.43h on the same device, which is performed for the four parameter combinations of the extreme values of ξ_0 and ξ_1 and for the midpoint of the parameter space $(\xi_0, \xi_1) = (120°\mathrm{C}, 50\%)$. In consequence, 120 solution evaluations for an iterative optimization would take 291.6h on average with the FEM – more than 500 times slower than the PINN optimization, without taking into account the additional necessary calculation time for the Jacobian and Hessian matrices. Small cost function adaptions would require another time-intensive iterative optimization with the FEM, while the PINN model can be reused. Thereby, the mean of the relative error (Eq. 16) of the PINN solution evaluated at the 5678 FEM nodes at 251 equidistant time steps amounts to less than 0.16% for the temperature and 0.14% for the curing degree.

4. Conclusions

This work presents a manufacturing process optimization in dependence on process and material properties using data-free surrogate models of physics-informed neural networks (PINNs). At the beginning, the surrogate modeling of PINNs is demonstrated by solving the parameterized PDEs of a 1D heat transfer problem with up to four parameters. In a second example of a more complex thermochemical curing process of a composite plate, the surrogate model approach is applied in a cost optimization process of the cured plate in dependence on one process and one design parameter. Additionally, the solution of the thermochemical manufacturing process is analyzed with respect to the parameters, demonstrating how PINN surrogate models can support scientific investigations of manufacturing and design processes.

The solution of the PINNs for the 1D in-stationary heat transfer problem as well as the thermochemical curing process are in good agreement with the FDM and FEM computations. Furthermore, by appending parameters to the input of PINNs, surrogate models are trained without labeled data. The surrogate models of the 1D problem are able to produce accurate results in the tested scenarios for a wide range of different initial and boundary conditions, defined by four parameters. In the in-stationary 2D case, the surrogate models produce accurate results for widely varied heating temperatures and fiber volume contents.

The curing process is analyzed in dependence on those two parameters, supported by the PINN surrogate models. Automatic differentiation is applied on the surrogate model to determine derivatives, specific parts of the PDE and other quantities, helping to understand the influence of the parameters on the manufacturing process. It is described, how the heating temperature of the process and the fiber volume content of the composite influence the occurrence of hot spots and the temperature distribution in the considered curing process.

The surrogate model is integrated in an iterative optimizer to find the minimum plate manufacturing cost, corresponding to a simple cost model and under the constraint that a curing degree of more then 99% has to be achieved in the whole plate. The optimum is calculated rapidly by the optimizer, due to the fast evaluations of the PINNs surrogate model. The optimal solution is predicted accurately, since it is indeed located in the area of low plate costs.

Overall, the results of this work show that PINNs are a promising option for efficient product developments. However, further development is necessary for application to real-world developments. These fall into three categories: Firstly, inclusion of additional process parameters in order to consider more complex manufacturing scenarios or introduce geometry parameters to support part design. One example application of high relevance for design could be a stiffening rib of variable dimensions (e.g., wall thickness, depth, draft angle). A PINN-based surrogate could then allow to optimize the rib-design in short time given different material properties and processes conditions. Secondly, improvement of model reusability e.g. by means of transfer learning. Thirdly, increase training efficiency to combat shortcoming of PINNs, e.g. adequately resolving of high-frequency solution components.

Data availability

The authors do not have permission to share data.

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.

Acknowledgment

This work is part of the IGF research project OptiFeed (21949 N) of the research association Forschungskuratorium Textil e.V., Reinhardtstraße 14–16, 10117 Berlin, funded via the AiF within the program for supporting the "Industrial Collective Research" (IGF) from funds of the Federal Ministry of Economic Affairs and Climate Protection (BMWK). It has also been funded by the LuFo research project Electra (20W1912D) regarding cost-efficient composite manufacturing scenarios. The financial support by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) is gratefully acknowledged. This work is also part of the DFG AI Research Unit 5339 funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 459291153, regarding the combination of physics-based simulation with AI-based methodologies for the fast maturation of manufacturing processes.

Appendix A

A.1. Neural network and training settings

In Table 1 the neural network and training settings of the PINNs results in Section 3 are listed. In this work, an Adam optimizer with a mini-batch sampling is used to train the PINNs, which is a common approach in the PINNs literature [29]. The PINNs are implemented with the SciANN library, using Keras and TensorFlow backends [30]. Additionally, a dimensionless approach is applied for the PINNs analogously to the implementation in [10], due to its superiority regarding robustness and numerical stability. However, for readability, the examples in this work are described with dimensioned variables. In the training, a training sample contains one value for each neural network input. The distribution of training samples defines, how many of the training samples are located on the domain respectively the individual boundaries. E.g. for a distribution (D/IC/BC-LRTB) of 50/25/12.5 %, the training samples are located randomly to 50% on the domain (D), to 25% on the initial condition (IC) and to 6.25% each on the left, right, top and bottom boundary condition (BC-LRTB). The learning rate of the training is exponentially decreased (ED).

In the thermochemical curing process of Section 3.2, a sequential training algorithm is implemented to train the temperature and curing degree PINN, like proposed in [8]. The algorithm starts with the training of the temperature by holding the curing degree

constant zero in the heat equation. Then, the gained PINN solution of the temperature is fixed to train the curing degree α network. Subsequently, the training of the individual networks alternates for four additional rounds. The training settings in Table 1 of the 2D thermochemical problem refer to the training of one individual training of the temperature respectively curing degree PINN.

References

[1] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S.G. Johnson, Physics-informed neural networks with hard constraints for inverse design, SIAM J. Sci. Comput. 43 (6) (2021) B1105–B1132.

[2] C. Zimmerling, C. Poppe, O. Stein, L. Kärger, Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning, Mater. Des. 214 (2022) 110423.

[3] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, J. Beyerer, Optimisation of manufacturing process parameters using deep neural networks as surrogate models, Procedia CiRP 72 (2018) 426–431.

[4] P. Jiang, Q. Zhou, X. Shao, Surrogate model-based engineering design and optimization, Springer, 2020.

[5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, arXiv preprint arXiv:1711.10561.

[6] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Comput. Methods Appl. Mech. Eng. 379 (2021), https://doi.org/10.1016/j.cma.2021.113741. URL:https://www.sciencedirect.com/science/article/pii/S0045782521000773 113741.

[7] 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.

[8] S. Amini Niaki, E. Haghighat, T. Campbell, A. Poursartip, R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture (2021). doi: 10.1016/j.cma.2021.113959. URL: https://www.sciencedirect.com/science/article/pii/S0045782521002966.

[9] B. Moseley, A. Markham, T. Nissen-Meyer, Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations, arXiv preprint arXiv:2107.07871.

[10] E. Haghighat, D. Amini, R. Juanes, Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training, Comput. Methods Appl. Mech. Eng. 397 (2022) 115141.

[11] E. Hosseini, P.G. Ghanbari, O. Müller, R. Molinaro, S. Mishra, Single-track thermal analysis of laser powder bed fusion process: Parametric solution through physics-informed neural networks, Comput. Methods Appl. Mech. Eng. 410 (2023) 116019.

[12] S. Liao, T. Xue, J. Jeong, S. Webster, K. Ehmann, J. Cao, Hybrid full-field thermal characterization of additive manufacturing processes using physics-informed neural networks with data, arXiv preprint arXiv:2206.07756.

[13] M. Penwarden, S. Zhe, A. Narayan, R.M. Kirby, Physics-informed neural networks (pinns) for parameterized pdes: A metalearning approach (2021). doi:10.48550/ARXIV.2110.13361. URL: https://arxiv.org/abs/2110.13361.

[14] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 394 (2019) 136–152.

[15] L. Lu, P. Jin, G.E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv preprint arXiv:1910.03193.

[16] Optimization and root finding (scipy.optimize) x2014; SciPy v1.9.1 Manual — docs.scipy.org,https://docs.scipy.org/doc/scipy/reference/optimize.html#module-scipy.optimize, [accessed 22-Sep-2022].

[17] J.R. Cannon, The one-dimensional heat equation, no. 23, Cambridge University Press, 1984.

[18] A. Johnston, P. Hubert, G. Fernlund, R. Vaziri, A. Poursartip, Process modeling of composite structures employing a virtual autoclave concept, Sci. Eng. Compos. Mater. 5 (3–4) (1996) 235–252, https://doi.org/10.1515/SECM.1996.5.3-4.235 [cited 2022–09-06] URL: doi: 10.1515/SECM.1996.5.3-4.235.

[19] M. Kamal, Integrated thermorheological analysis of the cure of thermosets, SPE Tech. Pap. 19 (1973) 187–191.

[20] A. Bernath, L. Kärger, F. Henning, Accurate cure modeling for isothermal processing of fast curing epoxy resins, Polymers 8 (11). doi:10.3390/polym8110390. URL:https://www.mdpi.com/2073-4360/8/11/390.

[21] A. Bernath, F. Groh, W. Exner, C. Hühne, F. Henning, Experimental and numerical study of the spring-in of angled brackets manufactured using different resins and fiber textiles, J. Compos. Mater. 53 (28–30) (2019) 4173–4188, https://doi.org/10.1177/0021998319855423.

[22] R.A. Witik, F. Gaille, R. Teuscher, H. Ringwald, V. Michaud, J.-A.E. Månson, Economic and environmental assessment of alternative production methods for composite aircraft components, J. Clean. Prod. 29 (2012) 91–102.

[23] MatWeb – The Online Materials Information Resource — matweb.com, https://matweb.com/search/DataSheet.aspx?MatGUID=3f64b985402445c0a5af911135909344&ckck=1, [Accessed 23-Sep-2022].

[24] MatWeb – The Online Materials Information Resource — matweb.com, https://matweb.com/search/DataSheet.aspx?MatGUID=3f64b985402445c0a5af911135909344, [accessed 23-Sep-2022].

[25] MatWeb – The Online Materials Information Resource — matweb.com, https://matweb.com/search/DataSheet.aspx?MatGUID=bf4a3a511f334316a0daccf8d43bfaa8, [accessed 23-Sep-2022].

[26] MatWeb – The Online Materials Information Resource — matweb.com, https://matweb.com/search/DataSheet.aspx?MatGUID=d924cb03eaca4d46ad9db2c0bb32c1b1, [accessed 23-Sep-2022].

[27] K. Prasad, R. Kramer, N. Marsh, M. Nyden, T. Ohlemiller, M. Zammarano, Numerical simulation of fire spread on polyurethane foam slabs, in: Fire research division, BFRL, NIST annual fire conference, Citeseer, 2009.

[28] L.M. Chiacchiarelli, I. Puri, D. Puglia, J.M. Kenny, L. Torre, Cure kinetics of a highly reactive silica-polyurethane nanocomposite, Thermochim. Acta 549 (2012) 172–178, https://doi.org/10.1016/j.tca.2012.09.031. URL:https://www.sciencedirect.com/science/article/pii/S0040603112004662.

[29] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, arXiv preprint arXiv:2201.05624.

[30] E. Haghighat, 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.