# ACCELERATED GRADIENT-BASED DESIGN OPTIMIZATION VIA DIFFERENTIABLE PHYSICS-INFORMED NEURAL OPERATOR: A COMPOSITES AUTOCLAVE PROCESSING CASE STUDY

#### A PREPRINT

# Janak M. Patel\*

Applied Research, Quantiphi Marlborough, MA 01752, USA janak.patel@quantiphi.com

#### **Anirudh Deodhar**

Applied Research, Quantiphi Marlborough, MA 01752, USA anirudh.deodhar@quantiphi.com

#### Milad Ramezankhani

Applied Research, Quantiphi Marlborough, MA 01752, USA milad.ramezankhani@quantiphi.com

#### **Dagnachew Birru**

Applied Research, Quantiphi Marlborough, MA 01752, USA dagnachew.birru@quantiphi.com

February 18, 2025

## ABSTRACT

Simulation and optimization are crucial for advancing the engineering design of complex systems and processes. Traditional optimization methods require substantial computational time and effort due to their reliance on resource-intensive simulations, such as finite element analysis, and the complexity of rigorous optimization algorithms. Data-agnostic AI-based surrogate models, such as Physics-Informed Neural Operators (PINOs), offer a promising alternative to these conventional simulations, providing drastically reduced inference time, unparalleled data efficiency, and zero-shot super-resolution capability. However, the predictive accuracy of these models is often constrained to small, low-dimensional design spaces or systems with relatively simple dynamics. To address this, we introduce a novel Physics-Informed DeepONet (PIDON) architecture, which extends the capabilities of conventional neural operators to effectively model the nonlinear behavior of complex engineering systems across high-dimensional design spaces and a wide range of dynamic design configurations. This new architecture outperforms existing state-of-the-art models, enabling better predictions across broader design spaces. Leveraging PIDON's differentiability, we integrate a gradient-based optimization approach using the Adam optimizer to efficiently determine optimal design variables. This forms an end-to-end gradient-based optimization framework that accelerates the design process while enhancing scalability and efficiency. We demonstrate the effectiveness of this framework in the optimization of aerospace-grade composites curing processes achieving a 3× speedup in obtaining optimal design variables compared to gradient-free methods. Beyond composites processing, the proposed model has the potential to be used as a scalable and efficient optimization tool for broader applications in advanced engineering and digital twin systems.

 $\textbf{\textit{Keywords}} \ \ \text{Neural operator} \cdot \text{Physics-informed DeepONet} \cdot \text{Gradient-based optimization} \cdot \text{Composite materials} \cdot \text{Curing processes}$ 

# 1 Introduction

Simulation and optimization play a pivotal role in modern engineering, serving as essential tools for analyzing, designing, and enhancing complex systems and processes. Engineers leverage simulation models to explore design spaces, evaluate

<sup>\*</sup>Corresponding author.

performance, and guide decision-making through optimization. Traditionally, numerical simulation methods, such as finite element analysis, have been widely used in combination with optimization algorithms to determine optimal design variables [29, 4, 30]. However, these methods are computationally expensive and time-intensive, making them less practical for real-time or iterative-based design processes [38]. To address these challenges, machine learning-based surrogate models have been developed as alternatives to reduce computational costs [20, 31, 37]. Once trained, these models provide fast predictions, enabling near real-time optimization. However, surrogate models often require large datasets for training, which is prohibitive in many industrial scenarios with data scarcity, resulting in unreliable or physically implausible predictions.

Recently, Physics-Informed Neural Networks (PINNs) are introduced [22], which incorporate governing partial differential equations (PDEs) into the loss function of a neural network. This allows the model to learn directly from the underlying physics, combining the strengths of numerical methods with machine learning-based surrogate modeling. While PINNs offer a promising alternative for solving PDEs without requiring labeled data, they are typically trained on fixed initial and boundary conditions (ICs/BCs) [34]. This limitation makes them less practical for iterative design optimization tasks, where system configurations (e.g., different BCs) frequently change. In contrast, neural operators such as DeepONet [16] and Fourier neural operator (FNO) [14], which directly map infinite-dimensional input-output function spaces on bounded domains, are capable of learning a family of PDEs rather than a single instance [2, 11]. Inspired by PINNs, Physics-Informed Neural Operators (PINOs) extend this capability by incorporating physics constraints into the loss function, allowing them to solve PDE families without relying on data [36, 6]. Unlike PINNs, PINOs enable real-time predictions under varying conditions, making them ideal for design optimization. Despite the advantages of PINOs in capturing system dynamics under varying conditions, existing architectures may struggle with high-dimensional and large design spaces for real-world engineering problems. Additionally, traditional gradient-free optimization methods, such as Particle Swarm Optimization (PSO) [18] and Genetic Algorithms (GA) [27], often require a large number of function evaluations, making them computationally expensive and inefficient for high-dimensional optimization [1].

To address these challenges, we propose an accelerated gradient-based optimization framework tailored for advanced engineering design. Specifically, Our work focuses on composite materials processing, where achieving an optimal cure cycle is critical for ensuring material quality. Composite materials are extensively used in aerospace, automotive, and marine industries due to their high strength, durability, and lightweight properties [13]. These materials undergo a polymerization process governed by a temperature and pressure cycle known as the cure cycle [28]. The cure cycle must be optimized to ensure uniform resin curing while minimizing residual stress and deformation [8]. Since changes in part geometry or material properties often require adjustments to the cure cycle, optimization becomes essential for ensuring high-quality and reliable composites.

Numerous studies have utilized PINO models to simulate the thermochemical evolution of composites during autoclave curing. For instance, a Physics-guided Neural Operator is developed for composite manufacturing [3]. Similarly, physics-informed FNO was applied to analyze the curing process of carbon-fiber composites [17]. Furthermore, a Physics-Informed DeepONet, enhanced with advanced features such as nonlinear decoders and curriculum learning is developed for modeling complex thermochemical processes [23]. However, the predictive performance of these models remains satisfactory only when applied to single-variable designs, limited design spaces, or systems with low behavioral complexity. This limitation may arise from the inherent constraints of the original neural operator architecture in handling nonlinear and stiff problems, as well as complex physics-informed loss landscape [12] and neural network's spectral bias [21]. To address these challenges and ensure that the developed surrogate model remains accurate when exposed to high-dimensional and large design spaces for optimization tasks, we propose an advanced Physics Informed DeepONet (PIDON) architecture. In particular, it incorporates domain decomposition with separate DeepONets allocated to each temporal subdomain, as well as input coordinate normalization to mitigate spectral bias [19]. Furthermore, each subdomain is equipped with a nonlinear decoder to better capture the complex dynamics of the PDE. The proposed PIDON model serves as an efficient and accurate surrogate for inverse design, enabling near real-time spatiotemporal predictions for given design conditions. This capability enhances design space exploration and accelerates the identification of optimal design variables to achieve desired material properties. Leveraging the differentiability of PIDON [15], we utilize a gradient-based optimization technique, which significantly reduces the number of function evaluations and improves scalability for larger design spaces compared to gradient-free methods [1]. The Adam optimizer [10] is used to obtain the optimal design variables. We benchmark this framework against gradient-free methods, including PSO and GA. The main contributions of this paper are summarized as:

- Design and implement an improved PIDON model for composites manufacturing, enabling an accurate and accelerated exploration of high-dimensional and large design spaces.
- Develop an end-to-end AI-driven design optimization framework enabling  $3\times$  speedup over conventional gradient-free approaches.

# 2 Methodology

## 2.1 Composites autoclave processing

The autoclave processing in the manufacture of composites involves placing resin-impregnated fibers and a tool into an autoclave, where the system undergoes a cure cycle to achieve the desired material properties, as illustrated in Figure 1.a. The temperature distribution within the part  $(T_c)$  and tooling  $(T_t)$ , along with the progression of the resin's degree of cure (DOC), are critical state variables in composites systems. The thermochemical behavior of a 1D composite-tool system in an autoclave is described by the following one-dimensional anisotropic heat conduction equation:

$$\frac{\partial T_t}{\partial t} = a_t \frac{\partial^2 T_t}{\partial z^2}, \quad z \in [0, L_t] 
\frac{\partial T_c}{\partial t} = a_c \frac{\partial^2 T_c}{\partial z^2} + b_c \frac{\partial \alpha}{\partial t}, \quad z \in [L_t, L_c]$$
(1)

Here, T is the temperature,  $\alpha$  is DOC,  $L_c$  is the material length,  $L_t$  is the tool length, and t and t are the spatiotemporal coordinates. The subscripts t, t, and t represent the tool, composite part, and resin, respectively. The parameter t denotes the thermal diffusivity, and t represents the heat generation coefficient. For the curing process of a composite system with thermoset resin, the cure rate is governed by the resin's cure kinetics, which are typically represented by an ordinary differential equation. For the AS4/8552 epoxy resin system, it can be expressed as follows[8]:

$$\frac{\partial \alpha}{\partial t} = A \exp\left(-\frac{\Delta E}{RT}\right) \frac{1}{1 + \exp\left(C(\alpha - (C_0 + C_T T))\right)} \alpha^m (1 - \alpha)^n \tag{2}$$

where,  $\Delta E$  represents the activation energy, R is the gas constant, and  $C_0$ ,  $C_T$ , m, n, and A are experimentally determined constants and parameter. The values used for this study can be found in [9]. Considering the convective heat transfer between the autoclave air  $T_a$  and the composite system, the boundary conditions are governed by:

$$(T_a - T_c|_{z=L_c}) = \frac{k_c}{h_{\text{top}}} \frac{\partial T_c}{\partial z} \Big|_{z=L_c}$$

$$(T_t|_{z=0} - T_a) = \frac{k_t}{h_{\text{bot}}} \frac{\partial T_t}{\partial z} \Big|_{z=0}$$
(3)

Here,  $h_{\text{top}}$  and  $h_{\text{bot}}$  are the convective heat transfer coefficients (HTCs) on the top and bottom surfaces of the compositetool system, respectively, while  $k_c$  and  $k_t$  represent the thermal conductivity of the composite and tool, respectively. The initial temperature of the part is typically assumed to be uniform throughout. For this study, we assume an initial temperature of 20°C. The initial DOC is assumed to be either zero or a very small value for an uncured part and in this study, is set to 0.05. The part thickness is fixed for this study, while the tool thickness is included as one of the design variables. To manage inconsistencies in the total system length and interface location resulting from variations in tool thickness, local coordinates are introduced, following the approach outlined by [23].

A typical two-hold cure cycle consists of two distinct stages where the air temperature is held constant at specific levels for predetermined durations. This approach enables a controlled progression of the curing process, allowing optimal material properties to be achieved while minimizing potential issues such as under-curing or over-curing [5]. The cure cycle is defined using six key design variables: heating rates  $r_1$  and  $r_2$  (the rate at which the air temperature inside the autoclave increases), hold durations  $hd_1$  and  $hd_2$  (the time periods for which the temperature is maintained constant at the respective hold temperatures) and hold temperatures  $ht_1$  and  $ht_2$  (the target temperatures held during the first and second stages).

#### 2.2 Differentiable Simulator: Physics Informed DeepONet

The architecture of DeepONet has two primary components: a branch and a trunk network [16]. The branch network takes as input the sensor point evaluations  $u=[u(x_1),u(x_2),\ldots,u(x_m)]$  and generates a feature representation  $b=[b_1,b_2,\ldots,b_q]^T\in\mathbb{R}^q$  as its output. Similarly, the trunk network encodes the spatiotemporal coordinates of the PDE system y into a feature embedding  $t=[t_1,t_2,\ldots,t_q]^T\in\mathbb{R}^q$ , with the same dimensionality as the output of the branch network. The outputs from these networks are combined using an element-wise product operation, followed by summation, to produce the final output of the DeepONet:

$$G_{\theta}(u)(y) = \sum_{k=1}^{q} b_k t_k + b_0 \tag{4}$$

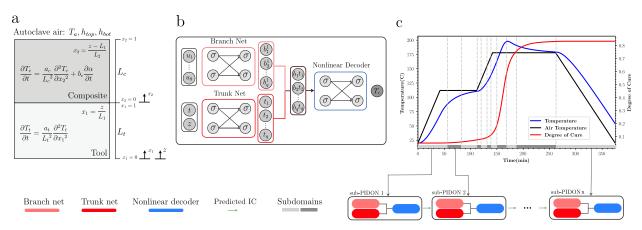


Figure 1: a) Schematic representation of the composite-tool system inside an autoclave, including local coordinates  $x_1$  and  $x_2$  (adopted from [23]). b) Architecture of the proposed sub-PIDON model for predicting part temperature  $G^{T_c}$ . The same architecture is utilized for other output variables, including DOC  $G^{\alpha}$  and tool temperature  $G^{T_t}$ . c) Illustration of the proposed PIDON framework with a nonlinear decoder and domain decomposition, designed for thermochemical analysis during the composite curing process.

Data-driven DeepONet often requires large amounts of training data, which can be difficult to access in many real-world engineering applications. The physics-informed variant of DeepONet, namely PIDON, integrates governing equations directly into the loss function as regularizers, eliminating DeepONet's reliance on data. Specifically, the output of the PIDON is constrained to satisfy the governing equations by minimizing the loss function:

$$\mathcal{L}(\theta) = \mathcal{L}_{IC}(\theta) + \mathcal{L}_{BC}(\theta) + \mathcal{L}_{physics}(\theta)$$
(5)

where  $\mathcal{L}_{IC}$ ,  $\mathcal{L}_{BC}$ , and  $\mathcal{L}_{physics}$  represent the initial condition (IC) loss, boundary condition (BC) loss, and physics loss, respectively. Assuming a constant initial condition and a Robin boundary condition in the introduced composites case study,  $\mathcal{L}_{IC}$  and  $\mathcal{L}_{BC}$  can be expressed as:

$$\mathcal{L}_{IC}(\theta) = \frac{1}{NQ_{ic}} \sum_{i=1}^{N} \sum_{j=1}^{Q_{ic}} \left| G_{\theta}(u^{(i)})(y_j^{(i)}) - s^{(i)}(y_j^{(i)}) \right|^2$$
 (6)

$$\mathcal{L}_{BC}(\theta) = \frac{1}{NQ_{bc}} \sum_{i=1}^{N} \sum_{j=1}^{Q_{bc}} \left| \alpha G_{\theta}(u^{(i)})(y_j^{(i)}) + \beta \nabla G_{\theta}(u^{(i)})(y_j^{(i)}) - \gamma \right|^2$$
(7)

Here,  $u^{(i)}$  denotes the i-th input function,  $y_j^{(i)}$  represents the j-th collocation point , and  $G_{\theta}$  is the output of the PIDON. The term  $s^{(i)}(y_j^{(i)})$  corresponds to the solution of the partial differential equation (PDE) at  $y_j^{(i)}$ , conditioned on the i-th input function. For the Robin boundary condition,  $\alpha$ ,  $\beta$ , and  $\gamma$  are non-zero constants determined by the physical characteristics of the problem. The physics loss  $\mathcal{L}_{physics}$  is defined as:

$$\mathcal{L}_{physics}(\theta) = \frac{1}{NQ} \sum_{i=1}^{N} \sum_{j=1}^{Q} \left| \mathcal{N}(u^{(i)}(x), G_{\theta}(u^{(i)})(y_j^{(i)})) \right|^2$$
 (8)

Here,  $\mathcal{N}$  denotes the nonlinear differential operator. The parameter N represents the number of distinct input function combinations sampled from the design space, whereas Q indicates the number of residual points employed to enforce the physical constraints. These N and Q are hyperparameters that can be tuned to balance the model's performance and computational efficiency. DeepONet is considered differentiable because it is a neural network-based approach that learns a solution operator, and a neural network is a differentiable function [16].

## 2.2.1 Proposed Architecture of PIDON

We introduce a novel PIDON architecture designed to model the highly nonlinear dynamics of composite-tool systems during the curing process. The key components of the framework are outlined below.

Network architecture. This architecture incorporates a branch network, a trunk network, and a nonlinear decoder, drawing inspiration from advancements in operator learning [25, 7, 23]. As illustrated in Figure 1.b, the branch network is designed with multi-input functionality, accommodating 9 input parameters: six cure cycle parameters  $(r_1, r_2, hd_1, hd_2, ht_1, ht_2)$ , two equipment design parameters  $(h_{top}, h_{bot}, representing the convective heat transfer coefficients on the top and bottom surfaces of the composite-tool system, respectively), and one tool design parameter (tool thickness, <math>L_t$ ). The trunk network takes spatio-temporal coordinates t and t as inputs. Standard DeepONet architectures with linear decoders require a large output dimension for branch and trunk networks to model nonlinear systems, making them computationally expensive and ineffective in such scenarios. In this work, we incorporate a nonlinear decoder by introducing an additional neural network that takes as input the combined output of the branch and trunk networks and generates the final output of PIDON.

Temporal domain decomposition. The composites curing process is a time-consuming process which requires solving the corresponding PDE equations over a long temporal domain. This poses some training challenges due to spectral bias and training difficulties such as activation saturation from large input coordinates. A temporal domain decomposition strategy is proposed, where a single DeepONet is trained with initial conditions as additional input functions [35]. This enables the model to independently learn solution operators for different subdomains while ensuring temporal continuity. However, inaccuracies in predicting initial conditions can introduce and propagate errors in the model predictions. Our framework improves temporal domain decomposition by employing multiple DeepONets with separate spatiotemporal input normalization for each subdomain. In particular, the temporal domain is divided into *n* smaller temporal subdomains based on the system's physical characteristics, with each subdomain modeled by an independent DeepONet, termed sub-PIDON (Figure 1.c). This setup enables each sub-PIDON to accurately capture the dynamics specific to its respective subdomain while reducing errors associated with relying on a single model to represent both initial conditions and other input functions within the system. The incorporation of separate subdomain normalization alongside domain decomposition effectively mitigates spectral bias over extended temporal domains by ensuring that the solution frequency encountered within each subdomain remains low [19].

Multi-output functionality. The PIDON framework predicts three key state variables for the thermochemical evolution in the composites curing process:  $T_c$ ,  $T_t$ , and  $\alpha$ . Due to the system's complexity, a single DeepONet may not effectively learn all these variables. Instead, we develop three dedicated and decoupled neural operators:  $G^{T_c}$  for composite temperature,  $G^{T_t}$  for tool temperature, and  $G^{\alpha}$  for the DOC, with each DeepONet specifically designed for its respective variable while maintaining consistent architectural principles. Thus, for each subdomain (highlighted with light and dark grey in Figure 1.c), we train three sub-PIDON, each dedicated to an output variable.

#### 2.2.2 Training Procedure

As depicted in Figure 1.c, the proposed PIDON framework is trained sequentially across multiple temporal subdomains. The training process begins with the initialization and training of the first sub-PIDON module (sub-PIDON 1) using the system's global IC across the design space (i.e., input functions). Upon completion of its training, this module generates predictions that serve as the local IC for the subsequent sub-PIDON module (sub-PIDON 2). In this manner, the IC for each subdomain is determined based on the predictions of the preceding sub-PIDON model, as indicated by the green arrows in Figure 1.c. This iterative process continues until all sub-PIDON modules corresponding to the defined subdomains are fully trained. The subdomains are initially partitioned into segments of equal width; however, their widths are adaptively refined during training. Specifically, if the total training loss for a given subdomain fails to reach a predefined threshold, the subdomain is further subdivided into two smaller intervals, thereby simplifying the learning task for the corresponding sub-PIDON modules. This adaptive approach enables more substantial progression in regions of the time domain where nonlinearity is less pronounced, while ensuring that highly complex regions are modeled with finer resolution. As a result, the framework maintains both computational efficiency and predictive accuracy across varying levels of system complexity.

# 2.3 AI-Driven Accelerated Design Optimization Framework

We developed an AI-driven accelerated design optimization framework for composite synthesis in autoclaves, as illustrated in Figure 2. This framework integrates a generalized, accurate, and computationally efficient PIDON model, serving as a differentiable simulator to optimize the curing process. By leveraging the zero-shot super-resolution capabilities of PIDON across diverse design parameters and its differentiability for gradient-based optimizers like Adam, our framework enables rapid and efficient exploration of design variables.

## 2.3.1 Design Tasks

Several studies have focused on optimizing the cure cycle for composite synthesis to achieve various objectives. For instance, one study aimed to minimize residual stress and ensure uniform curing [26], while another focused on maximizing the DOC, controlling peak temperature, minimizing post-gelation gradients, and reducing curing time [33].

This study aims to optimize the cure cycle of AS4/8552 composites by balancing mechanical performance and structural integrity through four key objectives. The first objective is achieving a desired DOC within the range  $0.85 < \alpha\big|_{t=t} < 0.95$ , ensuring optimal mechanical properties while avoiding brittleness [24]. Second, minimizing DoC gradients  $\frac{\partial \alpha}{\partial x}\big|_{t=t}$  is critical for reducing residual stress and shrinkage [37]. Third, maximum part temperature (exotherm) must remain below 185 °C to prevent thermal degradation [5]. Finally, controlling thermal lag ( $\Delta T < 20$  °C) ensures even curing by limiting temperature differences between autoclave air and composite parts [5]. These objectives are essential to optimize curing for performance and material integrity.

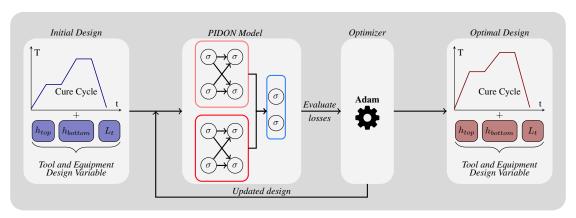


Figure 2: An accelerated gradient-based design optimization framework: Using the PIDON model, the framework predicts the spatio-temporal evolution of part temperature and DOC, calculates the loss, and iteratively adjusts design variables through gradient-based optimization to find the optimal design.

#### 2.3.2 Problem Formulation

As elaborated in subsection 2.1, we consider nine design variables (i.e., six parameters related to the cure cycle and three associated with tool/equipment design) represented as:  $u = [r_1, r_2, hd_1, hd_2, ht_1, ht_2, h_{top}, h_{bot}, L_t]$ . The optimization problem is formulated to minimize a total loss function  $\mathcal{L}(u)$ , composed of four individual loss terms representing the objectives defined in the above subsection:

$$\mathcal{L}(u) = \mathcal{L}_1(u) + \mathcal{L}_2(u) + \mathcal{L}_3(u) + \mathcal{L}_4(u) \tag{9}$$

Using the PIDON model as a surrogate, we predict part temperature and DOC, used in the loss calculations. For the first objective, a penalty function is defined based on the desired DOC. A penalty function ensures the DOC remains within the desired range of 0.85 to 0.95. If the DOC falls outside this range: either the minimum of DOC across the laminate is below 0.85 or the maximum of DOC across the laminate exceeds 0.95, a penalty is applied; otherwise, the loss is zero. This can be expressed as:

$$\mathcal{L}_{1}(u) = \begin{cases} |0.85 - \min(G_{\theta}^{\alpha}(u)(y_{t}))|, & \text{if } \min(G_{\theta}^{\alpha}(u)(y_{t})) < 0.85; \\ |\max(G_{\theta}^{\alpha}(u)(y_{t})) - 0.95|, & \text{if } \max(G_{\theta}^{\alpha}(u)(y_{t})) > 0.95; \\ 0, & \text{else.} \end{cases}$$
(10)

Here,  $y_t = [t|_{t=t}, z]$  and z is the collocation points in the spatial domain z. A second objective is to minimize the gradient of the DOC across the laminate at the end of the curing process. Since the PIDON model is differentiable, the gradient is computed using automatic differentiation across the laminate thickness, and it is then averaged. This can be expressed as follows:

$$\mathcal{L}_2(u) = \frac{1}{N_z} \sum_{i=1}^{N_z} \left| \frac{\partial (G_\theta^\alpha(u)(y_t^{(i)}))}{\partial z} \right| \tag{11}$$

Here,  $N_z$  is the number of collocation points along the z-coordinate. The third objective ensures that the maximum part temperature does not exceed 185°C. Hence, a penalty function is introduced as follows:

$$\mathcal{L}_{3}(u) = \begin{cases} \left| 185 - \max(G_{\theta}^{T_{t}}(u)(y)) \right|, & \text{if } \max(G_{\theta}^{T_{t}}(u)(y)) > 185; \\ 0, & \text{else.} \end{cases}$$
(12)

The fourth objective aims to limit the thermal lag to values below 20°C. This objective can be expressed as follows:

$$\mathcal{L}_{4}(\phi) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \left| 20 - (T_{a}(y^{(i)}) - G_{\theta}^{T_{t}}(\phi)(y^{(i)})) \right|, \\ \text{if } (T_{a}(y^{(i)}) - G_{\theta}^{T_{t}}(\phi)(y^{(i)})) > 20; \\ 0, \text{else.} \end{cases}$$
(13)

# 2.3.3 Optimizer

The optimization problem for the inverse design of composites autoclave processing is formulated as: as follows:

$$u^* = \arg\min_{u} \mathcal{L}(u),$$
 subject to:  $u_{\min} \le u \le u_{\max}.$  (14)

The bounds  $u_{\min}$  and  $u_{\max}$  define the feasible design space for the design variables u and are summarized in Table 4. They are typically defined by the inherent physical constraints of the process, such as the range of HTCs and temperature achievable by the autoclave oven. These constraints also align with the ranges of the input functions used during the training of the PIDON model. By adhering to these bounds, the design variables remain within the range of values on which the PIDON model was trained, ensuring it provides accurate and reliable predictions. The solution  $u^*$  represents the optimal set of design parameters that minimizes the total loss L(u), while satisfying all objectives, constraints, and bounds.

To solve this optimization problem, we employ a gradient-based approach, leveraging the differentiability of the PIDON model. Gradient-based methods are particularly advantageous for differentiable objective functions, as they use gradient information to efficiently explore the design space [1, 32]. In this case, the loss function depends on the output of PIDON, which itself is a function of the input (i.e., design variables). By leveraging automatic differentiation, we can compute the gradient of the loss with respect to the design variables by propagating the gradients through the PIDON using the chain rule. These methods converge with fewer function evaluations, making them computationally efficient compared to sampling-based or heuristic approaches [1]. Specifically, we apply the Adam optimizer [10], which adapts the learning rate during optimization, improving the convergence speed and stability.

The optimization process involves the following steps (Figure 2):

- 1. **Initial Guess:** Select the initial design parameters  $u^0$ , either randomly or based on domain knowledge, and pass them to the PIDON model.
- 2. **Forward Prediction:** Use the PIDON model to predict the spatio-temporal evolution of part temperature and DOC across the laminate based on the current design variables.
- 3. Loss Evaluation: Compute the individual loss functions based on the predicted temperature and DOC values.
- 4. **Gradient Computation:** Utilize automatic differentiation to compute the gradients of the total loss function  $\mathcal{L}(u)$  with respect to the design parameters u. This step involves back-propagating the gradients through the PIDON model.
- 5. Update Design Variables: Apply a gradient-based optimizer (e.g., Adam) to update the design variables u in the direction that minimizes the loss function using the computed gradients.

Design variables are clipped within the specified bounds presented in Table 4 to prevent erroneous predictions. If a variable exceeds the upper limit, it is clipped to the upper value; if it falls below the lower bound, it is clipped to the lower limit. Each loss function is normalized to maintain consistent scales and prevent dominance of any objective. To compare efficiency, we implemented Nesterov-Adam (NAdam), PSO, and GA, using the same PIDON model and loss calculation procedure as the Adam optimizer for consistency.

#### 3 Results and Discussion

#### 3.1 Validation of PIDON Model

To train the PIDON model, the time domain was divided into 11 subdomains with various lengths, accommodating the level of complexity within each subdomain. All PIDON models used an identical architecture, as detailed in Table 3.

Table 1: Comparison of our proposed PIDON (ours), PINO [23], and FNO [17] based on mean absolute error(MAE) and maximum error (MAX) for temperature and DOC predictions.

Variable	PIDON (ours)	PINO	FNO	
variable	MAE   MAX	MAE   MAX	MAE   MAX	
$T_c(^{\circ}C)$	0.189   0.94	0.362   2.231	0.226   3.2	
$\alpha$	0.001   0.008	0.002   0.02	0.008   0.03	

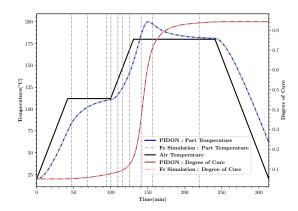


Figure 3: Comparison of predicted part temperature and DOC from the PIDON model with FE simulation at the midpoint of the composite.

We trained the model with 600 random input parameter combinations (i.e., input functions) spanning a wide range of values (Table 4). We evaluated the model against finite element simulations as ground truth across five different test cases. Specifically, the part temperature and degree of cure (DOC) at the midpoint of the composite for a given set of input functions are compared with FE simulation results, as illustrated in Figure 3. The plot demonstrates that the PIDON model provides highly accurate predictions, with its results closely overlapping those of the FE simulations. We also benchmarked its performance against previously developed operator-based models for composites processing, namely the Physics-Informed Neural Operator (PINO) [23] and the physics-informed Fourier Neural Operator (FNO) [17] (Table 1). We used mean absolute error (MAE) and mean maximum error as the evaluation metrics to compare model predictions with finite element simulations. The MAE was computed by averaging the absolute errors within and across all test cases. The mean maximum error was obtained by averaging the maximum errors from each test case. The results clearly show that the PIDON outperforms both PINO and FNO models, achieving MAE and mean maximum error that are approximately 50% lower than those reported for the other models, despite being trained on a wider range of input functions. This demonstrates the superior accuracy and reliability of the proposed PIDON approach for predicting temperature and DOC in composite systems.

#### 3.2 Design Optimization via Differentiable Neural Operator

A trained PIDON model is used as a surrogate model to identify the optimal design parameters for the autoclave curing process. Losses are evaluated using a  $20\times100$  grid of collocation points, with 20 points in the spatial domain (z) and 100 in the time domain. The performance of the framework was evaluated for two composite part thicknesses: 20 mm, as discussed in this section, and 30 mm, which is detailed in Appendix B. Figure 4 shows the key outcomes of the optimization procedure for a random initial state  $u^0 = [2.5, 1.5, 56, 117, 112, 176, 75, 100, 2.2]$ . Figure 4.a shows the evolution of the total loss during each iteration of the optimization process using the Adam optimizer. Figure 4.b compares the initial and optimized cure cycle profiles, including the part temperature behavior at the midpoint of the composite. Although the air temperature is higher for the optimized design compared to the initial design, the maximum part temperature remains below the threshold in both cases. The initial and optimal DOC distributions across the laminate thickness are compared in Figure 4.c. In the initial design, the DOC remains way below the 0.85 threshold, while the optimized design achieves a DOC exceeding 0.85 across the entire laminate thickness. These results clearly demonstrate the effectiveness of the proposed framework in achieving substantial improvements in cure cycle performance. The

Table 2: Performance comparison between gradient-based (i.e., Adam and NAdam) and gradient-free (i.e., PSO and GA) optimization models, highlighting computational time, PIDON Model calls and optimization objective metrics. Gradient-based optimizer results are averaged over 10 initial guesses, while gradient-free results use a single run. Model calls indicate the total number (forward + backward) function calls per optimization iteration.

Optimizer	Time (Min.)	Model Calls (per iteration)	Mean DOC Gradient	Max. $T_c$ (=<185)	Mean Thermal Lag (=<20)	Mean DOC (>=0.85)
Adam	20.37	1+2	0.0044±0.0002	185.21±0.25	14.10±0.67	0.853±0.0002
NAdam	20.02	1+2	0.0043±0.0001	185.12±0.16	14.16±0.61	0.852±0.00008
PSO	58.7	$10 \times (1+1)$	0.0057	185.18	14.11	0.847
GA	69.03	$  100 \times (1+1)  $	0.0045	185.07	14.91	0.852

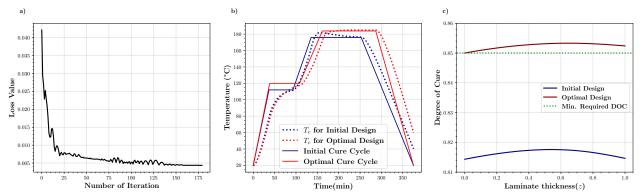


Figure 4: a) Evolution of the total loss during each iteration of the optimization process using the Adam optimizer. b) Comparison of the initial and optimized cure cycle profiles using the Adam optimizer, including the part temperature  $(T_c)$  behavior at the midpoint. c) DOC evolution across the laminate thickness from the initial (blue) to optimized (red) design variables. All results correspond to the optimization process with an initial guess for the design parameters  $u^0 = [2.5, 1.5, 56, 117, 112, 176, 75, 100, 2.2]$ .

optimization framework yields the optimized design variables  $u^* = [2.61, 1.2, 70, 125, 119, 183.9, 52.68, 91.61, 2.75]$  which satisfies all objectives, ensuring a high-quality manufactured part. The total computation time for a single optimization run is 20.37 minutes.

The obtained optimal design aligns very well with the theoretical expectations. Specifically, a higher value of  $r_1$  and a lower value of  $r_2$  ensures efficient heating during the first ramp phase while preventing excessive heat buildup (caused by resin polymerization) during the second phase. The optimal design has the highest  $ht_1$  to maximize heat during the first hold, ensuring a cure rate above 0.85 while avoiding excess heat in the second hold. For  $L_t$ , the optimal design has a mid-range value. While a higher tool thickness can help mitigate excessive exothermic reactions, it may lead to a non-uniform DOC across the laminate, thus the chosen thickness can balance these effects.

Further, to demonstrate the robustness of our framework, we present the optimization results using Adam for 10 different initial guesses. The average objective evolution across these runs is shown in Figure 5. The figure illustrates the progression of key objectives during optimization, including the average DOC across the laminate (Figure 5.a), maximum part temperature (Figure 5.b), average axial cure gradient (Figure 5.c), and average thermal lag (Figure 5.d). The blue line represents the mean value across the 10 initial guesses, while the shaded band indicates the standard deviation. From the figure, it is evident that in some initial designs, the DOC is below 0.85. However, the optimal designs consistently achieve a DOC exceeding 0.85 across the laminate thickness. Additionally, the variation in DOC across the thickness is significantly reduced in the optimized designs compared to the initial designs. This improvement is further reflected in the steady decrease of the DOC gradient with each optimization step. Since DOC and maximum part temperature are correlated, initial designs with higher DOC exhibit higher part temperatures. In the optimization process, the maximum part temperature, which initially varies widely, stabilizes precisely at 185°C in the optimized designs. Lastly, the average thermal lag successfully remains below 20°C throughout the optimization procedure.

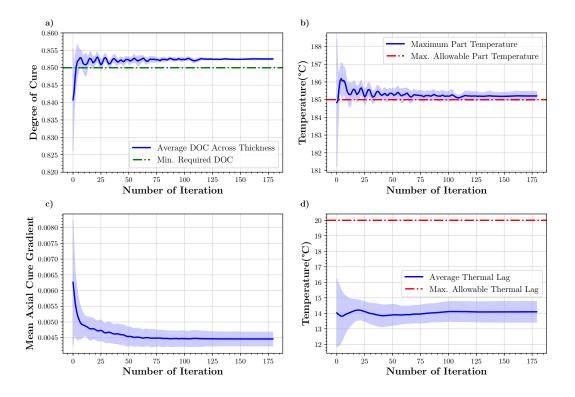


Figure 5: Evolution of key metrics during optimization: a) Average DOC across the laminate thickness, b) Maximum Temperature observed in the laminate, c) Average Axial Cure Gradient, and d) Average Thermal Lag, at each optimization step. Results are averaged over 10 different initial guesses, with shaded bands representing ± standard deviation to indicate variability across the optimization runs.

## 3.3 Comparison to gradient-free methods

In addition to our proposed optimization framework, we implemented and compared alternative optimization techniques, including NAdam, PSO, and GA, to evaluate their performance. The NAdam was executed for 180 iterations. The PSO was implemented with a swarm of 20 particles, running for 25 iterations, while the GA utilized a population size of 100 and was executed for 100 iterations. The performance of these optimizers, along with their computation times, is summarized in Table 2. For gradient-based optimizers, such as Adam and NAdam, the results are averaged over 10 different initial guesses to assess robustness. For population-based optimizers, since they are less sensitive to initial guesses, a single optimization run was executed and reported. In terms of the achieved optimal states, the gradient-based and gradient-free methods resulted in comparable performance. In terms of the number of function calls during the optimization process, gradient-based methods call the PIDON model once for forward prediction and twice for backward propagation (axial DOC gradient calculation and optimizer gradient update) per iteration (1 + 2). In contrast, gradient-free methods require one forward and one backward call for axial DOC gradient calculation for each individual in the population per iteration  $(1+1) \times p$ , where p is the population size. This makes gradient-based optimizers significantly more efficient with fewer function evaluations during the optimization process. Hence, the gradient-based optimizers demonstrated a significant computational advantage, converging approximately three times faster than their gradient-free counterparts. This clearly indicates the advantage of utilizing a differentiable neural operator model in conjunction with a gradient-based optimization approach for accelerated and robust design optimization in advanced manufacturing and materials processing applications.

# 4 Conclusions

In this work, we present an end-to-end accelerated design optimization framework for advanced engineering design, specifically developed for the autoclave curing process in composite materials processing. At the core of the framework

is an enhanced PIDON model, specifically designed to capture the nonlinear dynamics of the autoclave process with high accuracy. The improved PIDON integrates a nonlinear decoder and temporal domain decomposition techniques, delivering superior performance compared to existing neural operator-based models across wide range of diverse input functions and extended temporal domains. By leveraging the differentiability of the trained PIDON model, the framework utilizes the gradient-based Adam optimizer to efficiently identify optimal design parameters. When benchmarked against gradient-free methods such as PSO and GA, the Adam optimizer achieves comparable performance while operating approximately three times faster. This enables the accelerated and accurate identification of optimal design states for the autoclave curing process. Furthermore, the framework is highly adaptable and can be extended to address other advanced engineering design challenges, making it well-suited for digital twin applications.

# References

- [1] Kelsey R Allen, Tatiana Lopez-Guevara, Kimberly Stachenfeld, Alvaro Sanchez-Gonzalez, Peter Battaglia, Jessica Hamrick, and Tobias Pfaff. Physical design using differentiable learned simulators. *arXiv preprint arXiv*:2202.00728, 2022.
- [2] Nicolas Boullé and Alex Townsend. A mathematical guide to operator learning. *arXiv preprint arXiv:2312.14688*, 2023.
- [3] Gengxiang Chen, Yingguang Li, Xu Liu, Charyar Mehdi-Souzani, Qinglu Meng, Jing Zhou, and Xiaozhong Hao. Physics-guided neural operator for data-driven composites manufacturing process modelling. *Journal of Manufacturing Systems*, 70:217–229, 2023.
- [4] Dilmurat Dolkun, Weidong Zhu, Qiang Xu, and Yinglin Ke. Optimization of cure profile for thick composite parts based on finite element analysis and genetic algorithm. *Journal of Composite Materials*, 52(28):3885–3894, 2018.
- [5] Janna Noemi Fabris. A framework for formalizing science based composites manufacturing practice. PhD thesis, University of British Columbia, 2018.
- [6] Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed deep neural operator networks. In *Machine Learning in Modeling and Simulation: Methods and Applications*, pages 219–254. Springer, 2023.
- [7] Ehsan Haghighat, Umair bin Waheed, and George Karniadakis. En-deeponet: An enrichment approach for enhancing the expressivity of neural operators with applications to seismology. *Computer Methods in Applied Mechanics and Engineering*, 420:116681, 2024.
- [8] Pascal Hubert, Andrew Johnston, Anoush Poursartip, and Karl Nelson. Cure kinetics and viscosity models for hexcel 8552 epoxy resin. In *International SAMPE symposium and exhibition*, pages 2341–2354. SAMPE; 1999, 2001.
- [9] Andrew A Johnston. An integrated model of the development of process-induced deformation in autoclave processing of composite structures. PhD thesis, University of British Columbia, 1997.
- [10] Diederik P Kingma. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [11] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [12] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.
- [13] Xiaoyang Li, Jihui Wang, Shuxin Li, and Anxin Ding. Cure-induced temperature gradient in laminated composite plate: Numerical simulation and experimental measurement. *Composite Structures*, 253:112822, 2020.
- [14] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [15] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3):1–27, 2024.
- [16] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint arXiv:1910.03193, 2019.

- [17] Qinglu Meng, Yingguang Li, Xu Liu, Gengxiang Chen, and Xiaozhong Hao. A novel physics-informed neural operator for thermochemical curing analysis of carbon-fibre-reinforced thermosetting composites. *Composite Structures*, 321:117197, 2023.
- [18] Lester James V. Miranda. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, 3, 2018.
- [19] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, 2023.
- [20] Julius Pfrommer, Clemens Zimmerling, Jinzhao Liu, Luise Kärger, Frank Henning, and Jürgen Beyerer. Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CiRP*, 72:426–431, 2018.
- [21] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- [22] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [23] Milad Ramezankhani, Anirudh Deodhar, Rishi Yash Parekh, and Dagnachew Birru. An advanced physics-informed neural operator for comprehensive design optimization of highly-nonlinear systems: An aerospace composites processing case study. *Engineering Applications of Artificial Intelligence*, 142:109886, 2025.
- [24] Florian Rothenhäusler and Holger Ruckdaeschel. Interplay of curing and thermal degradation in epoxy resins cured with amino acids: Influence of the maximum curing temperature on the network structure, crystal morphology and mechanical properties. *Journal of Applied Polymer Science*, 140(45):e54655, 2023.
- [25] Jacob Seidman, Georgios Kissas, Paris Perdikaris, and George J Pappas. Nomad: Nonlinear manifold decoders for operator learning. *Advances in Neural Information Processing Systems*, 35:5601–5613, 2022.
- [26] PH Shah, VA Halls, JQ Zheng, and RC Batra. Optimal cure cycle parameters for minimizing residual stresses in fiber-reinforced polymer composite laminates. *Journal of Composite Materials*, 52(6):773–792, 2018.
- [27] R. Solgi. geneticalgorithm: A python library for solving optimization problems using genetic algorithms, 2020.
- [28] A Brent Strong. Fundamentals of composites manufacturing: materials, methods and applications. Society of manufacturing engineer, 2008.
- [29] Giacomo Struzziero and Alexandros A Skordos. Multi-objective optimisation of the cure of thick components. *Composites Part A: Applied Science and Manufacturing*, 93:126–136, 2017.
- [30] Wenyuan Tang, Yingjie Xu, Xinyu Hui, and Wenchang Zhang. Multi-objective optimization of curing profile for autoclave processed composites: Simultaneous control of curing time and process-induced defects. *Polymers*, 14(14):2815, 2022.
- [31] KI Tifkitsis, TS Mesogitis, G Struzziero, and AA Skordos. Stochastic multi-objective optimisation of the cure process of thick laminates. *Composites Part A: Applied Science and Manufacturing*, 112:383–394, 2018.
- [32] Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 33:6111–6122, 2020.
- [33] Mehdy Vafayan, Mir Hamid Reza Ghoreishy, Hossein Abedini, and Mohammad Hossein Beheshty. Development of an optimized thermal cure cycle for a complex-shape composite part using a coupled finite element/genetic algorithm technique. *Iranian Polymer Journal*, 24:459–469, 2015.
- [34] Hengjie Wang, Robert Planas, Aparna Chandramowlishwaran, and Ramin Bostanabad. Mosaic flows: A transferable deep learning framework for solving pdes on unseen domains. *Computer Methods in Applied Mechanics and Engineering*, 389:114424, 2022.
- [35] Sifan Wang and Paris Perdikaris. Long-time integration of parametric evolution equations with physics-informed deeponets. *Journal of Computational Physics*, 475:111855, 2023.
- [36] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.
- [37] Zhenyi Yuan, Lingfei Kong, Dajing Gao, Xinxing Tong, Yu Feng, Guigeng Yang, Zhenchao Yang, and Shujuan Li. Multi-objective approach to optimize cure process for thick composite based on multi-field coupled model with rbf surrogate model. *Composites Communications*, 24:100671, 2021.

[38] Clemens Zimmerling, Christian Poppe, Oliver Stein, and Luise Kärger. Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Materials & Design*, 214:110423, 2022.

# A Hyper-parameters of PIDON and Optimizers

The hyperparameters for each sub-PIDON model are summarized in Table 3, including network architectures, optimizer settings, training parameters, and hardware specifications. Additionally, The range of input functions for which PIDON is trained is presented in Table 4.

For the Adam optimizer, a custom dynamic learning rate mechanism is implemented. If the loss does not improve for a specified number of iterations (defined by patience), the learning rate is reduced by the factor reduction\_factor, with a lower bound set by min\_lr. The optimizer starts with an initial learning rate of 0.01, performs up to 180 steps, and reduces the learning rate by a factor of 0.5 if no improvement is observed after 10 iterations, with the learning rate not dropping below  $1 \times 10^{-5}$ . A similar dynamic learning rate mechanism is employed for the Nadam optimizer. The GA used in this study was configured with a population size of 10, a mutation probability of 0.1, an elitism ratio of 0.01, and a crossover probability of 0.5. The selection process retained the top 30% of individuals as parents, applying uniform crossover for offspring generation. The PSO algorithm was set with a cognitive parameter of 0.5, a social parameter of 0.3, and an inertia weight of 0.9.

Table 3: Hyperparameters of the PIDON model.

ruote 3. Hyperpurume	ters or the rap or anouer	
Branch Network	[50, 50, 50]	
Trunk Network	[50, 50, 50, 50, 50]	
Non-linear Decoder	[50, 50, 50, 50]	
Optimizer	Adam	
Initial Learning Rate	$1 \times 10^{-3}$	
Learning Rate Decay	Decay rate of 0.9 every 1000 steps	
Training Epochs	200	
Training Library	JAX	
Hardware	NVIDIA T4 GPU with 104 GB of memory	

Table 4: Design variables (input functions) and their corresponding ranges.

Range
[1.2, 3]
[1.2, 3]
[50, 70]
[115, 125]
[100, 120]
[175, 185]
[70, 120]
[40, 90]
[2, 4]

# **B** Additional Results

To demonstrate the robustness of the proposed framework in addressing challenging scenarios, this section presents results for a 30 mm thick composite part. The optimization problem becomes significantly more challenging for the 30 mm thick material due to the excessive exothermic heat generation and the conflicting objectives of minimizing the maximum part temperature while achieving a sufficiently high DOC. The evolution of the total loss during optimization is shown in Figure 6.a, while Figure 6.b compares the initial and optimized cure cycles, along with the corresponding part temperatures at the composite midpoint. The initial cure cycle design exhibits excessive heat accumulation, resulting in a significantly elevated maximum part temperature. In contrast, the optimized design effectively regulates the maximum part temperature, bringing it to approximately 186.4°C—slightly above the target value of 185°C but still well within the acceptable range for practical applications [5]. The DOC distribution across the laminate thickness for both the initial and optimized designs is illustrated in Figure 7.a. In the initial design, the DOC remains below 0.85, whereas the optimized design achieves a DOC exceeding 0.85 across all regions, except those near the tool. Furthermore, the optimized design demonstrates a substantial reduction in DOC variation, as indicated by the flatter maroon line compared to the darker blue line, which is also reflected in the decreasing DOC gradient over successive iterations in the Figure 7.c. Figure 7.b depicts the progression of the maximum part temperature during optimization. Initially extremely high, the temperature progressively decreases and stabilizes at approximately 186.4°C in the final design. Similarly, the average thermal lag evolves as shown in Figure 7.d. Although it momentarily increases during

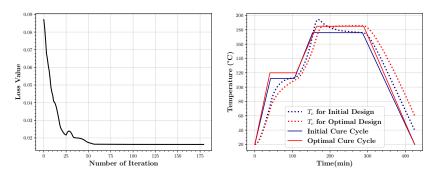


Figure 6: Optimization results for 30 mm thick composite part: a) Evolution of the total loss during each iteration of the optimization process using the Adam optimizer b) Comparison of the initial and optimized cure cycle profiles using the Adam optimizer, and the corresponding part temperature behavior at the midpoint.

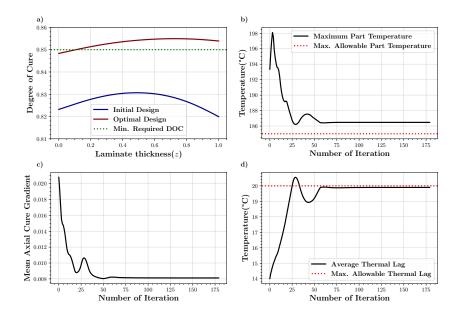


Figure 7: For 30mm thick material: a) DOC evolution across the laminate thickness from the initial (blue) to optimized (red) design variables. Evolution of b) maximum temperature (exotherm). c) average axial (through-thickness) cure gradient, and d) average thermal lag evolution at each optimization step.

intermediate steps, the thermal lag ultimately returns to below 20°C in the final optimized solution, aligning with acceptable design criteria. These findings underscore the framework's capability to handle complex, high-dimensional design problems with conflicting objectives, optimizing even for extreme and thick composite materials to yield feasible solutions that satisfy physical constraints and design objectives.

#### **B.1** Comparison to gradient-free methods

A comparative analysis of the gradient-based and gradient-free optimization approaches for the 30 mm thick composite material is presented in Table 2. The hyperparameters for each optimizer remain consistent with those used for the 20 mm thick material. Notably, gradient-based optimizers demonstrate a threefold speed advantage over their gradient-free counterparts. The optimization process for the 30 mm composite part poses significant challenges due to the conflicting nature of the objectives, including minimizing maximum part temperature and maximizing DOC. This is evident as gradient-free optimizers, such as GA and PSO, struggle to achieve a balanced solution. GA achieves a higher DOC but results in a maximum part temperature exceeding 185°C. Conversely, PSO maintains a maximum part temperature

Table 5: Performance comparison between gradient-based (i.e., Adam and NAdam) and gradient-free (i.e., PSO and GA) optimization methods, for a 30 mm thick composite part.

Optimizer	Computation Time (Min.)	Mean DOC (>=0.85)	Mean DOC Gradient	Max. T <sub>c</sub> (=<185)	Mean Thermal Lag (=<20)
Adam	19.7	0.853	0.0083	186.4	19.53
NAdam	19.25	0.853	0.0084	186.38	19.51
PSO	58.68	0.837	0.0098	184.98	19.11
GA	69.12	0.852	0.0091	188.87	15.02

below 185°C but fails to achieve a DOC above 0.85. In contrast, the gradient-based optimizers, Adam and NAdam, demonstrate an ability to effectively balance these competing objectives. While the maximum part temperature for both optimizers slightly exceeds the target value of 185°C, stabilizing at 186.4°C, this result is still within an acceptable range for practical purposes. Additionally, both optimizers achieve a DOC above 0.85, fulfilling the primary performance objectives. This highlights the capability of gradient-based methods in efficiently navigating complex, high-dimensional design spaces.