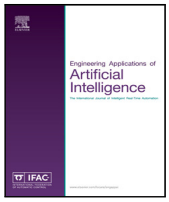
Engineering Applications of Artificial Intelligence 101 (2021) 104232

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications

Navid Zobeiry [a,*], Keith D. Humfeld [b]

[a] Materials Science & Engineering Department, University of Washington, Seattle, WA, United States of America
[b] DuoTech LLC., Federal Way, WA, United States of America

ARTICLE INFO

Keywords:
Physics-informed machine learning
Theory-guided feature engineering
Convective heat transfer
Advanced manufacturing
Industry 4.0

ABSTRACT

A physics-informed neural network is developed to solve conductive heat transfer partial differential equation (PDE), along with convective heat transfer PDEs as boundary conditions (BCs), in manufacturing and engineering applications where parts are heated in ovens. Since convective coefficients are typically unknown, current analysis approaches based on trial-and-error finite element (FE) simulations are slow. The loss function is defined based on errors to satisfy PDE, BCs and initial condition. An adaptive normalizing scheme is developed to reduce loss terms simultaneously. In addition, theory of heat transfer is used for feature engineering. The predictions for 1D and 2D cases are validated by comparing with FE results. While comparing with theory-agnostic ML methods, it is shown that only by using physics-informed activation functions, the heat transfer beyond the training zone can be accurately predicted. Trained models were successfully used for real-time evaluation of thermal responses of parts subjected to a wide range of convective BCs.

1. Introduction

In a given manufacturing process, a material typically undergoes several transformations (e.g. phase transformations in additive manufacturing of metallic or plastic parts, or convective curing of thermoset composites). From a pure mathematical point of view, the process can be represented using a series of governing Partial Differential Equations (PDEs) (Fernlund et al., 2018; Tlili et al., 2019; Zobeiry et al., 2016; Zobeiry and Duffner, 2018; Zobeiry and Poursartip, 2015). These PDEs are often derived based on conservation laws such as mass, momentum, and energy conservations throughout the processing (Mohseni et al., 2019; Sheikholeslami et al., 2019; Zobeiry and Duffner, 2018; Zobeiry and Humfeld, 2019). From an industrial perspective, it is essential to control the behavior of the material during processing to ensure end part quality (Fernlund et al., 2018; Zobeiry et al., 2020a, 2016). This is often achieved by precise control of Boundary Conditions (BCs) such as pressure and temperature histories during the fabrication process. The transformations of the material subjected to these BCs is usually studied prior to processing by solving the governing PDEs using numerical methods such as Finite Elements (FE) (Fabris et al., 2016, 2015, 2014; Johnston et al., 1996; Park et al., 2017). In addition to process simulation, in process measurements using thermocouples (TCs) or pressure sensors are often used. For example, TCs placed in critical locations are used for validation of numerical models or for feedback loops in an active control approach (Erol et al., 2016; Zobeiry et al., 2019a). However, considering the complexity of a fabrication process, controlling process parameters and precise measurement of material behavior during processing is not a trivial task. For example, in convective heating of parts in ovens, variation of airflow and consequently variation of heat transfer coefficients (i.e. unknown or variable BCs.), lead to formation of hot and cold spots. This in turn results in thermal gradients and thermal lag in the part (Fernlund et al., 2018; Mosavat et al., 2018; Park et al., 2017; Zobeiry et al., 2019a). For such a process, controlling the air temperature in the oven as a proxy to control the part temperature is quite challenging. In addition to uncertainty in fabrication process and unknown BCs, as the heat is transferred and then diffused to the center of the part via conduction, different zones in the material undergo different temperature histories. This is schematically shown in Fig. 1 where a part with a thickness of L is heated in a convective oven with asymmetric BCs (h_1 and h_2). As the air temperature is increased and then held at a constant temperature, the part temperature lags behind the air temperature. However, middle of the part and top and bottom surfaces undergo different temperature histories (Zobeiry et al., 2019a).

For such a complex problem, established industrial practices such as thermal profiling combined with process simulation using FE models are frequently employed by engineers for thermal management. However, FE models are not suitable to consider uncertainties in manufacturing processes including unknown BCs. As a result, engineers often rely on trial-and-error simulations based on different assumptions for BCs during processing. This combines with the fact that high fidelity

* Corresponding author.
E-mail address: navidz@uw.edu (N. Zobeiry).

https://doi.org/10.1016/j.engappai.2021.104232
Received 28 September 2020; Received in revised form 28 January 2021; Accepted 12 March 2021
Available online 25 March 2021

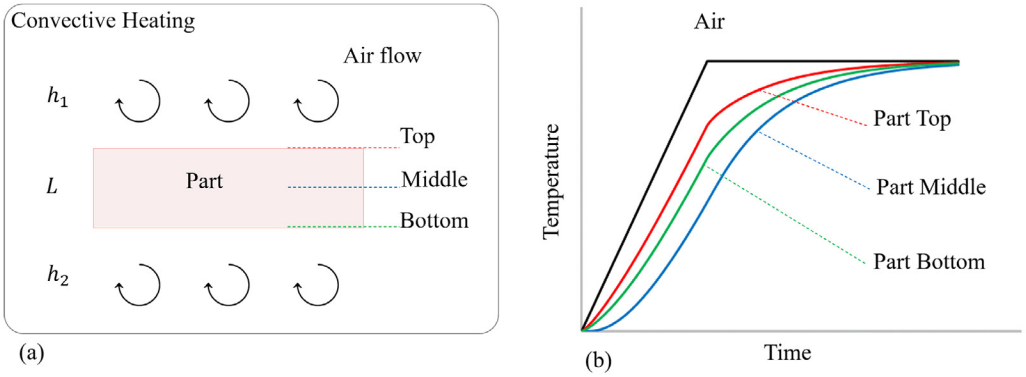


Fig. 1. (a) Convective heating of a part in an oven, and (b) Variation of temperature histories at different locations of the part.

FE tools are inherently slow, makes such an approach impractical for an industry 4.0 setting where fast and near real-time simulation capabilities are needed (Agrawal and Choudhary, 2016; Erol et al., 2016).

The rise of Machine Learning (ML) and AI in recent years offer an opportunity to develop fast surrogate ML models to replace traditional FE tools in manufacturing and general engineering applications. Several approaches have been explored in the literature in recent years:

- Theory agnostic ML models trained using physical data: Using limited physical data obtained from experiments or embedded sensors during manufacturing, predictive ML models have been developed. These have been used for different applications including turbulence modeling, and guiding assembly of aerospace parts (Manohar et al., 2018; Singh et al., 2017). The main challenge in these applications is developing ML techniques suitable for small physical datasets.
- Theory agnostic ML models trained using numerical data: By automating FE models to generate large numerical datasets, surrogate ML models have been trained in many applications. This includes turbulence modeling (Wang et al., 2017), simulation of manufacturing processes and failure in advanced composites (Zobeiry et al., 2019b, 2020c,b), heat transfer modeling (Barzegar Gerdroodbary, 2020), and stress analysis (Liang et al., 2018). One main challenge is the trade-off between fidelity and speed in FE tools, and consequently accuracy of the trained ML model. High-fidelity FE tools are inherently slow to set-up and perform which limits their applicability to generate large numerical datasets. On the other hand, low-fidelity FE tools are typically much faster and can be used to generate larger datasets. This generally comes at the cost of reduction in the model accuracy.
- Theory-guided ML models trained using a combination of governing physical laws, numerical data and physical data: It has been shown that by combining several data streams, and designing architecture and features of ML models based on governing physical laws, some limitations of previous approaches can be addressed (Karpatne et al., 2017; Wagner and Rondinelli, 2016; Zobeiry et al., 2020b). This includes using small physical datasets and low-fidelity FE tools to train ML models with acceptable accuracies even beyond their training zones (Zobeiry et al., 2019b, 2020c).
- Physics-informed ML models trained using governing PDEs: Instead of solving PDEs using FE tools to generate training datasets, in several studies governing PDEs have been used directly to train ML models (Han et al., 2018; Raissi, 2018; Raissi et al., 2019, 2017a,b; Sirignano and Spiliopoulos, 2018). In these studies, Physics-informed Neural Networks (PINNs) are typically trained by defining the loss function as the error in satisfying the governing PDE. By reducing the loss function, PINN is trained to ultimately satisfy the PDE. One key advantage of such an approach compared to previous approaches is that pre-generated training data such as FE results is not needed. In addition, once trained using BCs as inputs for the PINN, computational time will be much faster than FE models. This makes this approach quite suitable for implementation in industry 4.0 applications.

Following previous studies and established approaches to solve PDEs using ML (Han et al., 2018; Raissi, 2018; Raissi et al., 2019, 2017a,b; Sirignano and Spiliopoulos, 2018), a PINN is developed in this study to solve the heat transfer PDE with convective BCs in a representative manufacturing setting. A total loss function based on errors in satisfying heat transfer PDE, convective BC PDEs, and IC is defined. An adaptive normalizing scheme is developed capable of reducing errors in all loss terms simultaneously. In addition, physics-informed features and activation functions are defined based on the theory of heat transfer to accurately represent the underlying physics using the trained PINN. It is shown that using this approach, a PINN can predict heat transfer even outside its training zone. Comparison with FE results is used to validate the approach in 1D and 2D cases. Finally, the developed PINN model is used to successfully evaluate the thermal responses of parts subjected to various convective BCs in real-time. In an industry 4.0 setting, this leads to development of near real-time feedback control loops to adjust the process parameters and control the temperature history of the part.

2. Method

The general heat transfer equation for a given part can be written as (Incropera et al., 2007):

$$\frac{\partial}{\partial t}\left(\rho C_P T\right) = \frac{\partial}{\partial x}\left(k_{xx}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_{yy}\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_{zz}\frac{\partial T}{\partial z}\right) + \dot{Q} \tag{1}$$

In which T is the temperature, ρ is the part density, C_p is the part specific heat capacity, k is the part conductivity and $\dot{Q}$ is the rate of heat generation in the part. For sake of simplicity, the method is explained here for one-dimensional heat transfer (i.e. 1D). But this can be easily extended to higher dimensions which will be discussed later. The general heat equation can be simplified to represent the case of the one-dimensional heat transfer with no heat generation term:

$$\frac{\partial T}{\partial t} - \alpha\frac{\partial^2 T}{\partial x^2} = 0 \tag{2}$$

where α is the thermal diffusivity defined as:

$$\alpha = \frac{k}{\rho C_P} \tag{3}$$

The convective boundary condition is written as:

$$h\left(T_\infty - T_{boundary}\right) = k\left.\frac{\partial T}{\partial x}\right|_{boundary} \tag{4}$$

where h is the convective Heat Transfer Coefficient (i.e. BC), T_∞ is the air temperature around the part and $T_{boundary}$ is the part temperature at its surface.

Suppose that the prediction made by a neural network, $f\left(x,t,h_1,h_2\right)$, is intended to be a solution to the one-dimensional heat equation for any given boundary condition. The solution's adherence to the heat transfer PDE at any given point can be quantified as:

$$Error_{PDE} = \alpha \frac{\partial^2 f\left(x,t,h_1,h_2\right)}{\partial x^2} - \frac{\partial f\left(x,t,h_1,h_2\right)}{\partial t} \tag{5}$$

If the prediction made by the neural network is a perfectly trained solution to the one-dimensional heat equation, this error term will be zero at every point. For the two boundaries located at x_1 and x_2, the adherence to the convective boundary condition differential equations can be quantified as:

$$Error_{BC1} = -\left(T_\infty(t) - f\left(x_1,t,h_1,h_2\right)\right) + \frac{k}{h_1}\left.\frac{\partial f\left(x,t,h_1,h_2\right)}{\partial x}\right|_{x=x_1} \tag{6}$$

$$Error_{BC2} = \left(T_\infty(t) - f\left(x_2,t,h_1,h_2\right)\right) - \frac{k}{h_2}\left.\frac{\partial f\left(x,t,h_1,h_2\right)}{\partial x}\right|_{x=x_2}$$

If the prediction made by the neural network is a perfect solution to BCs, these error terms will be zero at any given time and for any given heat transfer coefficient. Heat transfer problems are frequently solved with an Initial Condition (IC), such as the one-dimensional solid being in thermal equilibrium at the beginning of the problem. One productive way to think of an IC is by recognizing that the IC simply describes a boundary condition applied at the time-dimension boundary $t = 0$. If the neural network intended to also represent a solution to such a boundary condition, the adherence to that boundary condition can be quantified as:

$$Error_{BC0} = T_\infty(0) - f\left(x,0,h_1,h_2\right) \tag{7}$$

For a collection of training data points, the loss term for training the neural network can be defined based on Eqs. (5)–(7):

$$Loss = Loss_{PDE} + \lambda_0 Loss_{BC0} + \lambda_1 Loss_{BC1} + \lambda_2 Loss_{BC2} \rightarrow \tag{8}$$

$$Loss = \frac{1}{N_{PDE}}\sum_{i=1}^{N_{PDE}} Error_{PDE}^2 + \frac{\lambda_0}{N_{BC0}}\sum_{i=1}^{N_{BC0}} Error_{BC0}^2 + \frac{\lambda_1}{N_{PDE}}\sum_{i=1}^{N_{BC1}} Error_{BC1}^2 + \frac{\lambda_2}{N_{BC2}}\sum_{i=1}^{N_{BC2}} Error_{BC2}^2$$

where λ values are scaling factors to normalize loss terms. Each term of the loss function is designed to calculate the mean square of the error term over the points for which that error term was evaluated. For a neural network that perfectly represents the solution to the one-dimensional heat equation, all of the loss values, evaluated over any arbitrary sets of points will sum to zero. Combining multiple loss functions such as these four loss functions into one cumulative loss function, however, faces a recognized challenge. That is the relative magnitudes of the losses impact the model training and result. If the magnitude of one loss, or its derivatives with respect to the model's weights, are significantly greater than others, the neural network will train to a solution that mainly minimizes one loss term. This issue is addressed later in the implantation section by developing an adaptive normalizing scheme that ensures the magnitudes of all losses are similar, resulting in minimizing all losses simultaneously during training.

3. Implementation and training of a physics-informed neural network

Training of a PINN to solve the heat transfer PDE with convective BCs was implemented in Python (V3.6.8), using Tensorflow and Keras libraries (V2.10). In this section, we discuss details of implementations including PINN architecture, theory-guided feature engineering, choice of activation function and adaptive normalization scheme. The training of PINN was based on selecting random (x,t,h) batches in each epoch and minimizing the loss function (Eq. (8)). This was achieved using built-in Keras optimizers to obtain weights and biases of the neural network to satisfy heat transfer PDE. Based on performing a grid search, Adam optimizer with a learning rate of 0.0001 was used with a batch size of 150 for each loss term. In most cases, an 100K epochs were used for training.

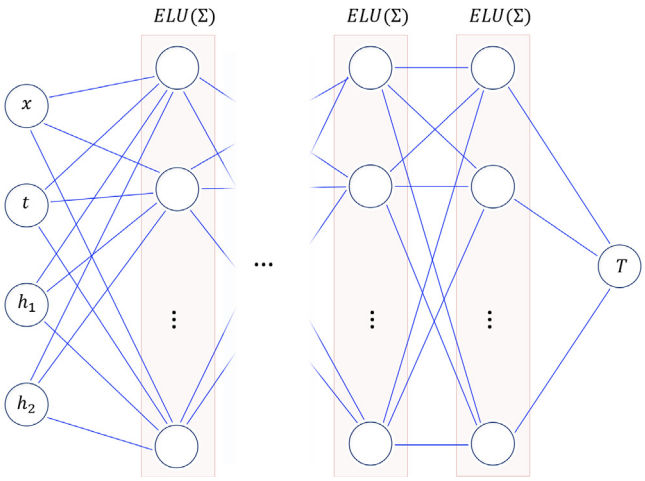


Fig. 2. Schematic of a neural network with a dense architecture to solve the heat transfer PDE. This is referred to as NN in this study.

3.1. Physics-informed NN architecture

Potentially, any neural network with a dense architecture can be trained to represent the solution to the heat transfer PDE. An example of such a network is schematically shown in Fig. 2. In addition to implementing physics-informed loss function, physics-informed feature engineering can be used as a complementary approach to better represent the governing physics. In the absence of heat generation and convection, the heat equation can be solved analytically by separation of variables (Incropera et al., 2007):

$$T(x,t) = T_0 + \left(T_{Max} - T_0\right)\sum_n A_n exp\left[-\frac{k}{\rho c_p}\left(\frac{n\pi}{L}\right)^2 t\right] cos\left(\frac{n\pi}{L}x\right) \tag{9}$$

where A_n are weights and n is an integer. In the presence of heat generation or convection, an analytic solution would take a different form if the problem were analytically solvable. That said, some of the underlying physics could be captured by engineering the features of the neural network to conform to the analytic form of the above solution of the heat equation. This is ignoring the effects of heat generation or convection.

To implement feature engineering, a first layer was made for the neural network by combining two pre-layers of terms as shown in Fig. 3. The first of the pre-layers took the position argument as an input, applied a trainable weighting and bias, and applied a sine function activation function. In the presence of a trainable bias, the sine and cosine functions are interchangeable. The second of the pre-layers took the time argument as an input, applied a trainable weighting and bias, and applied an exponential activation function. These two pre-layers were then multiplied on a 1:1 basis to create a layer with several terms of the form

$$exp\left[at + a_0\right] sin\left(bx + b_0\right) \tag{10}$$

The architectures shown in Figs. 2 and 3 are referred to as NN and PINN respectively in this study. The performance of these two architectures will be compared in the validation section. Using a grid search, 6 hidden layers with 32 nodes per layer was selected in both cases. In addition, 32 engineered features based on Eq. (10) were used in the PINN. It should be noted that different NN architectures have been proposed in the literature to solve PDEs. For example, Wang et al.

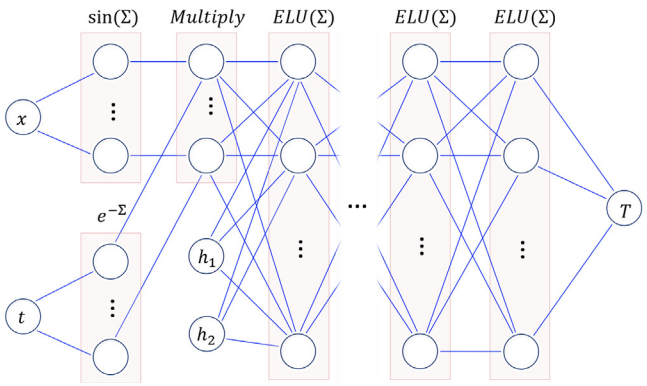


Fig. 3. Schematic of a neural network with physics-informed features to solve the heat transfer PDE. This is referred to as PINN in this study.

(2020) recently proposed a fully connected architecture with residual connections to improve the performance of physics-informed neural networks. This method has shown to be successful in matching a PDE solution with multiple oscillations. One advantage of the proposed PINN architecture in this study compared to other theory-agnostic methods is its capability to accurately capture the underlying physics. This is discussed later in the paper, and advantages are demonstrated for a case study.

3.2. Activation function

Choice of activation function has a significant impact on the success of training PINNs to solve PDEs. Common activation functions such as *sigmoid* or *tanh* may suffer from vanishing gradients problem (Pascanu et al., 2013). *ReLU* function on the other hand does not suffer from this problem and as a result widely used in many deep learning applications (Agarap, 2018):

$$ReLU\,(\Sigma) = \begin{cases} \Sigma & \Sigma \geq 0 \\ 0 & \Sigma < 0 \end{cases} \tag{11}$$

Several activation functions and their derivates are compared in Fig. 4. However, for solving a PDE such as heat transfer, it is necessary to take the first and second derivatives of the NN with respect to network inputs to calculate the loss function. As a result, training which is based on calculation of the derivate of the loss function, includes second and third derivates of the activation function. However, higher order derivatives of *ReLU* are equal to zero as shown in Fig. 4 for the second derivative. This makes the training process ineffective:

$$\frac{d^2 ReLU(\Sigma)}{d\Sigma^2} = 0 \tag{12}$$

It is necessary to train a PINN with an activation function that has a nonzero second derivative and allows effective deep network training. ELU activation function as shown in Fig. 4, for example, satisfies these requirements:

$$ELU\,(\Sigma) = \begin{cases} \Sigma & \Sigma \geq 0 \\ e^{\Sigma} - 1 & \Sigma < 0 \end{cases} \tag{13}$$

To accelerate training of physics-informed neural networks, as an alternative, adaptive activation functions have been proposed and validated in several applications (Jagtap and Karniadakis, 2019; Sun et al., 2020). For example, the above *ELU* function can be modified to an adaptive form as follows:

$$adaptive\_ELU\,(\Sigma) = \begin{cases} a\Sigma & \Sigma \geq 0 \\ e^{a\Sigma} - 1 & \Sigma < 0 \end{cases} \tag{14}$$

where a is a constant that is updated in each iteration of training based on the derivative of the loss function, and a pre-defined learning rate, η:

$$a^{i+1} = a^i - \eta \frac{d\,Loss}{da} \tag{15}$$

To compare different activation functions, three types of activation functions were investigated in this study:

- traditional activation functions such as *ReLU*, *ELU* and *tanh,* with a theory- agnostic NN architecture as shown in Fig. 2
- adaptive activation functions such as *adaptive_tanh* and *adaptive_ELU* with a theory- agnostic NN architecture as shown in Fig. 2
- physics-informed activation functions based on the governing heat transfer equations as described in previous section and shown in Fig. 3.

The results and comparison will be discussed in the next section for a one-dimensional heat transfer problem. It is shown that adaptive activation function converges faster to the solution. However, physics-informed activation is the only type that accurately represents the solution even outside of the training zone.

3.3. Adaptive normalization factors

The issue with relative magnitudes of loss values as described previously was resolved by using an adaptive normalization scheme. At the beginning of the training, the model is initialized using the Glorot uniform initializer in Keras (i.e. uniform distributions for weights and biases). Since this does not represent the solution to the system of equations, the relative magnitudes of the loss terms in Eq. (8) could be very different. This means that the largest error terms may dominate the training, such that error terms with smaller magnitudes would not be reduced as the training progresses. For a well-trained model however, loss terms are defined such that the individual error terms are comparable.

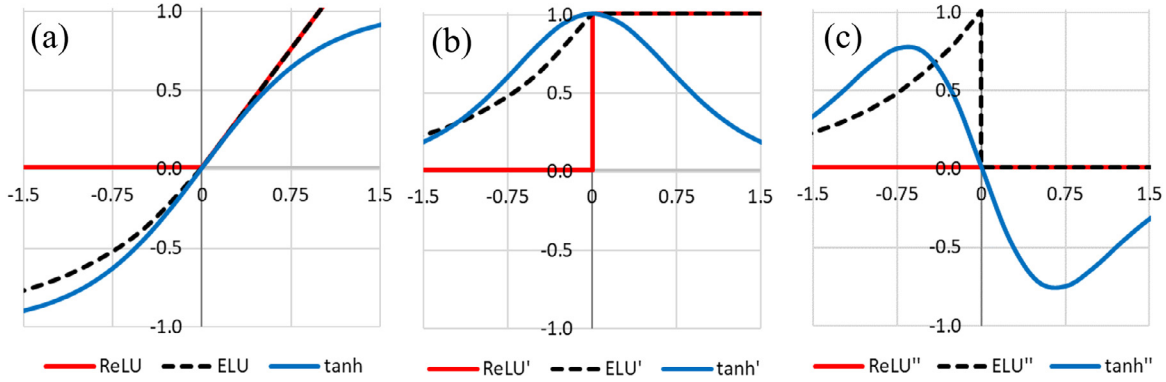


Fig. 4. (a) ReLU, ELU and tanh activation functions, (b) derivatives of activation functions, and (c) second derivatives of activation functions.

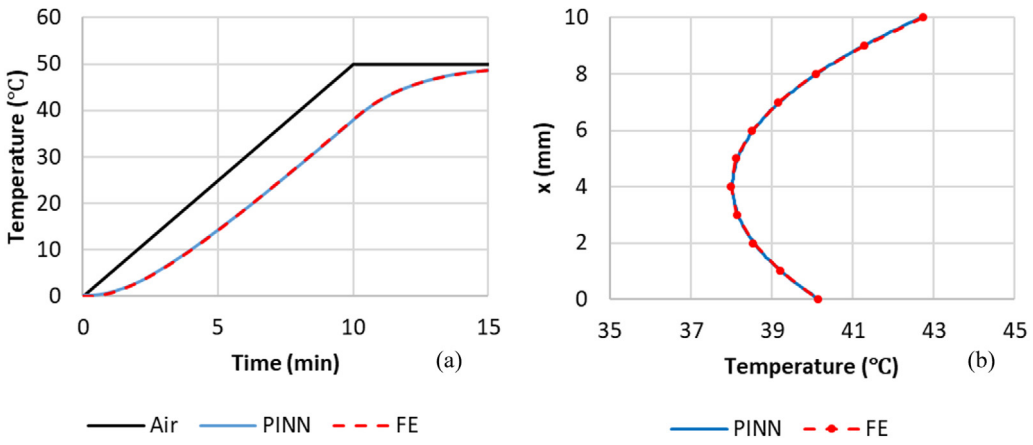


Fig. 5. (a) Temperature histories in the middle of a composite part subjected to an air temperature cycle, predicted using PINN and FE, and (b) Through-thickness temperature distribution after 10 min as predicted by PINN and FE.

The adaptive scheme was developed to update the normalization factors in Eq. (8) at regular intervals, e.g. every 100 epochs. New normalization factors were determined by evaluating the ratio between the specific loss term and the greatest loss term. If the ratio of loss terms was greater than some threshold (e.g. 0.01) then the normalization factor for the specific loss term was set to unity. If the ratio of loss terms was below that threshold, i.e. the loss term was much smaller than the greatest loss term, the normalization factor was set to the loss ratio divided by the threshold. This brought the relative error term to within the threshold for the current epoch.

As training continued, if the greatest loss term trained less slowly than this specific loss term, then the normalization factor would be intensified at the next update interval. This would put the model into a scenario where the model trains towards a solution for the maximum magnitude loss term with somewhat little influence from the other loss terms. While using this adaptive normalization scheme, however, this was not problematic. As the model continued to train, this maximum magnitude loss term was decreased faster than the other terms. The interplay between simple loss terms with rapidly decreasing normalization factors, and complex loss terms with larger normalization factors, resulted in normalization factors gradually approaching the threshold and set to unity. During some instances of model training, the adaptive scheme would set a normalization factor to unity in one update and then the loss ratio would drop down below the threshold during a later update. But with enough updates, all loss terms approach the same order of magnitude and all normalization factors became unity.

The proposed adaptive normalization factor in this paper was compared with the learning rate annealing algorithm proposed in the literature for physics-informed NNs (Wang et al., 2020). Although the annealing method has shown to be advantageous while using the SGD optimizer, in our study and using the adaptive normalization factor and Adam optimizer, no benefit was observed. In fact, the proposed method was faster in convergence while solving the heat transfer problem.

3.4. Prediction accuracy around boundary condition kinks

One of the difficulties observed in training neural networks is inaccurate predictions around kinks and discontinuities in the input data, for example in locations where $dT_{air}(t)/dt$ is not continuous (as shown in Fig. 1). Such discontinuity typically results in low training accuracy in the vicinity of that kink. As discussed above, one of the key advantages of using the differential equation and boundary conditions to train a neural network, was that no pre-generated training data needed to be defined Therefore, the specific points at which the model was trained did not need to be defined prior to training. This feature was exploited to improve the accuracy in the vicinity of the external air temperature kinks, by increasing the density of training points in the vicinity of these kinks. Likewise, the initial condition representing thermal equilibrium implies that there is a discontinuity in the derivative of the air temperature at the beginning of the problem. This led to inaccuracies in the predictions of the neural network for the lowest values of time. The accuracy was improved by increasing the density of training points near $t = 0$. For each discontinuity, in a zone equal to 10% of the normalized time around the kink, the density of the training points was increased. In this study, 50% of training points in each batch were selected around kinks and 50% away from the kinks.

To increase the accuracy of training, similar approaches have been explored in the literature. Most notably, Residual-based Adaptive Refinement (RAR) method has been proposed to increase accuracy in training of PINNs (Lu et al., 2019). In this approach, the areas with kinks (i.e. training zones with high errors) are automatically detected during training and added to the training points. Comparing to our proposed approach, RAR showed to be slower in training, and less accurate to solve the heat transfer problem. This was not surprising given that incorporating domain knowledge in training of ML models is highly advantageous. For cases where the locations of the kinks are not known, however, RAR approach is advantageous over random selection of training points.

4. 1D validation

The accuracy of the Physics-Informed Neural Network was investigated by comparing the predictions against calculations from a finite element model. The training and prediction accuracy of the PINN (Fig. 3) was also compared to the training and prediction of a simple NN (Fig. 2). Finally, to illustrate and validate the ability of the PINN to utilize the heat transfer coefficients on the boundaries as inputs, the trained PINN was run for a variety of heat transfer coefficients within the range of 10–100 W/m^2 K, and compared to the results of a series of matching finite element analyses. In all trainings and simulations, a carbon-fiber epoxy composite part with following nominal properties was used (Zobeiry et al., 2019a):

- $k = 0.47$ W/mK
- $\rho = 1573$ kg/m^3
- $C_p = 967$ J/kg K

An in-house developed FE code in python was used to solve the heat transfer PDE with convective BCs. Upon performing mesh size sensitivity, the geometry was discretized into 10 elements in each direction, and time was discretized into 5 s intervals.

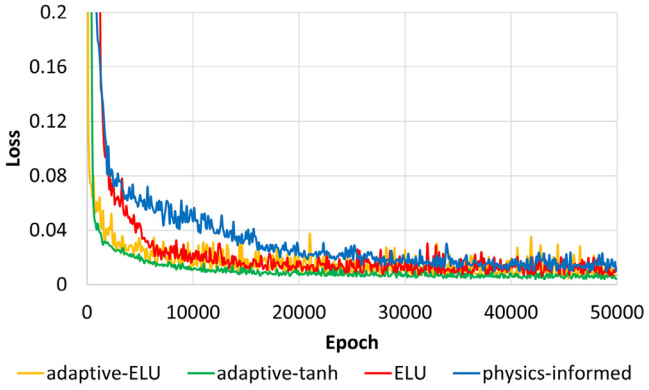


Fig. 6. Convergence of Loss functions during training using different activation functions and model architectures as shown in Fig. 2 (adaptive_ELU, adaptive_tanh and ELU) and Fig. 3 (physics-informed).

4.1. PINN versus FE

The Physics-Informed Neural Network described in Section 3 was trained to predict the time-dependent solution to the 1D heat equation in a composite part of 10 mm thickness. A uniform initial temperature boundary condition ($T_0 = 0$ °C), asymmetric convective heat transfer boundary conditions ($h_1 = 100$ W/m^2 K, $h_2 = 50$ W/m^2 K), and the specific external air temperature profile illustrated in Fig. 5(a) (a heating rate of 5 °C/min to 50 °C, and then held for 5 min) were considered. The prediction of the neural network for the temperature history in the middle of the part is shown by the solid blue line in Fig. 5(a). To validate the PINN predictions, a finite element simulation was run for an identical problem. The temperature in the middle of the part predicted by the finite element analysis is depicted by the dotted red line in Fig. 5(a). The prediction of the neural network matches the calculation of the finite element analysis quite well with a maximum deviation less than 0.97 °C at any given time. Fig. 5(b) illustrates the equivalence of the PINN predictions and the FE simulation results 10 min into the simulation, by showing the temperature through the thickness of the part. The predictions are asymmetric since heat transfer coefficients are different below and above the part. The finite element analysis, performed with ten elements through the thickness, can predict temperatures only at the nodes of the elements and must interpolate between those nodes, where the PINN can be evaluated at any point along the continuum, as the x-position of every point at which the PINN was trained was selected randomly from the continuum at each epoch.

4.2. PINN versus NN

Heat transfer in a 20 mm composite part with asymmetric convective BCs ($h_1 = 100$ W/m^2 K, $h_2 = 50$ W/m^2 K) and a 15-minute temperature cycle identical to previous case as depicted in Fig. 5(a) was modeled using three methods FE, NN and PINN. Several NNs were trained using *adaptive_ELU*, *adaptive_tanh* and *ELU* activation functions, and with simple architectures as shown in Fig. 2. Results were compared with a PINN trained with physics-informed functions as shown in Fig. 3 (PINN). All models were trained for the 15 min temperature cycle (i.e. $t\epsilon\,[0, 15\,\text{min}]$ during training). However, after training, they were used to predict the part temperature history 30 min beyond the training cycle (i.e. $t\epsilon\,[0, 45\,\text{min}]$ for prediction). NNs with adaptive activation functions were trained faster than the other NNs as shown in Fig. 6. After 5000 epochs the error in NN with *adaptive_tanh* reached the low value of 0.02. It took about 10,000 epochs for the NNs with *adaptive_ELU* and *ELU* to reach this loss value, and 25,000 epochs for the physics-informed NN. After 25,000 epochs, all networks reached a stable and equivalent total loss values (Fig. 6). The predictions of the networks after 50,000 training epochs for the center of the part are compared in Fig. 7 with FE predictions. Within the training zone, all networks predict a similar and accurate temperature history. Beyond the training zone, however, all solutions diverge from the FE predictions after a few minutes. Only the PINN with physics-informed activation functions can accurately capture the underlying physics beyond the training zone. This is due to the implementation of physics-informed functions in PINN to capture the underlying behavior of the heat transfer problem. At the time of 45 min, the difference between FE and PINN predictions reaches about 1.5 °C.

4.3. PINN with convective BCs as inputs

Using a PINN to predict the solution to the heat equation in the presence of convective heat transfer boundary conditions was made advantageous over using the finite element method. By extending the inputs of the neural network to include the heat transfer coefficients on both sides of the one-dimensional solid, as depicted in Fig. 3, large number of BCs can be analyzed in real-time. Increasing the dimensionality of the inputs of the neural network from 2 to 4 increased the number of points required to train the neural network, but this simply increased the number of epochs and the batch size required to train. A key advantage to using the physics informed neural network structure that calculates losses based on the adherence to the differential equations was that no training data needed to be generated prior to training the model. The PINN that included heat transfer coefficients as inputs was trained to similar accuracy as the two-input neural network. This

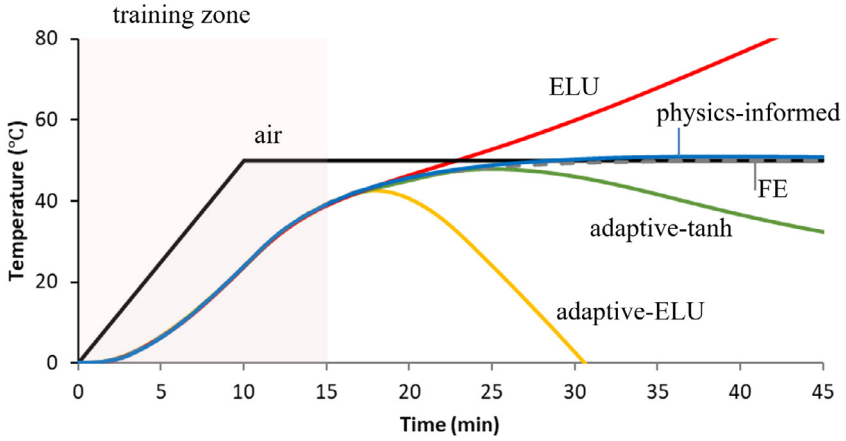


Fig. 7. Temperature histories at the middle of a part as predicted by FE, NN with adaptive_ELU, adaptive_tanh and ELU activation functions as shown in Fig. 2, and physics-informed activation function as shown in Fig. 3.

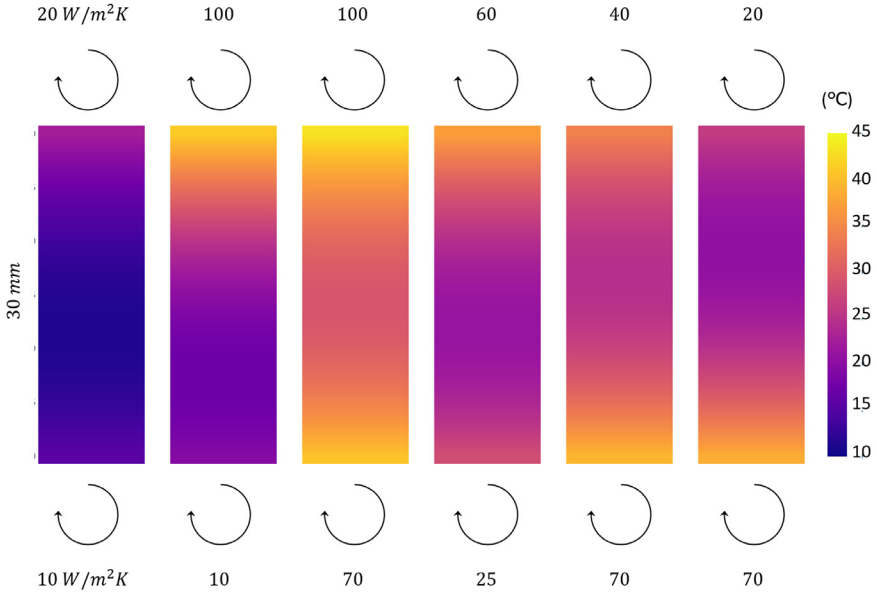


Fig. 8. Near real-time predictions of multiple heat transfer problems for a 30 mm composite part subjected to similar air temperature cycles but with different BCs using a trained PINN. Comparison to FE results showed a maximum error equal to 0.3 °C for all cases.

extended neural network enabled predictions of the temperature as a function of time, position, heat transfer coefficient above and heat transfer coefficient below.

Fig. 8 illustrates the ability of this neural network to accurately predict the temperature for a range of heat transfer coefficient combinations. The scenario illustrated in Fig. 8 is that of a 30 mm composite bar initially at 0 °C being heated by air that ramps linearly from 0 °C to 50 °C at a rate of 5 °C/min, and that then holds constant at 50 °C. The individual plots in Fig. 8 show the temperature across the length of the bar, at time = 15 min, for selected heat transfer coefficient combinations. PINN results were validated by comparison with FE results, which performed separately for each of the scenarios. Comparison to FE results showed a maximum error equal to 0.3 °C and an average error equal to 0.1 °C for all cases demonstrated in Fig. 8. If the same problem needed to be solved for different combinations of heat transfer coefficients, a new finite element analysis would need to be run for each such combination. The PINN that includes the heat transfer coefficient as inputs, however, is already trained to make predictions for any combination of heat transfer coefficients. Given that the neural network is already trained, the computational costs of making predictions for new combinations of heat transfer coefficients is very low.

5. Extension to 2D and beyond

The expansion of the PINN to two dimensions required the addition of one input (y-dimension) and the creation of another pre-layer for the y input. In addition, the error calculations for both heat transfer PDE and boundary PDE were adjusted to include derivatives with respect to y. The addition of another input caused the model to require more total data points to train. This resulted in the model requiring more epochs to train to an acceptable loss. Once the model was trained, the temperature at any point could be predicted by evaluating the model at the desired x, y, t, h_1, and h_2. Fig. 9 illustrates this capability by showing a heat map of the temperature throughout a 60 mm by 20 mm composite part, with heat transfer coefficients of 100 W/m^2 K on each the $x = 0$ and $y = 0$ boundaries, with the same air temperature profile illustrated in Fig. 5(a). The three individual illustrations within Fig. 9 show the temperature distributions at 5 min, 10 min and 15 min into the problem. The results were validated by comparison with FE results. This comparison showed a maximum error equal to 0.6 °C and an average error equal to 0.2 °C. Locations and values of maximum errors are identified in Fig. 9.

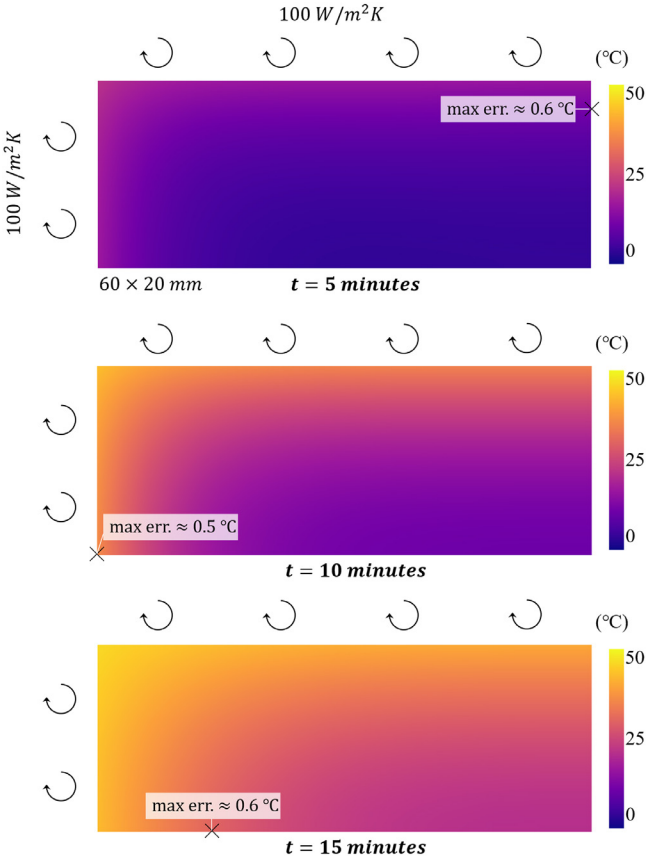


Fig. 9. Predictions of heat transfer in a 2D case using the trained PINN. Results are shown after 5, 10 and 15 min subjected to an air temperature in Fig. 5(a). Locations and values of maximum errors compared to FE results are identified for each time step.

5.1. Further extension

The expansion of the PINN to three dimensions would require the addition of one input (z-dimension), another pre-layer for the z input,

and adjustments of the error calculations. Additional heat transfer boundary conditions also require the addition of heat transfer inputs to the PINN. These additions will further increase the batch size and number of epochs required to train the model to an acceptable loss.

The neural network may also be modified to include the material properties as inputs. For the one-dimensional problem, it is sufficient to use the thermal diffusivity α and the thermal conductivity k as inputs. But to solve the problem in 2- and 3- dimensions the specific heat capacity (ρc_p) and thermal conductivity vector ($\vec{k} = (k_x, k_y, k_z)$) should be used. Adding two or three more inputs to the 2D or 3D neural network further increases the batch size and number of required epochs in training.

The air temperature profile used to train the model was explicitly captured within the training algorithm, necessitating a new model to be trained for each air temperature profile. The model could be expanded to include a set of parameters that define a family of air temperature profiles. For example, a family of air temperature profiles can be defined by adding one parameter for the initial temperature, one parameter for the heating rate, two parameters for the hold temperature and duration, one parameter for the cooling rate and a final parameter for the final temperature. This specific parameterization would require the addition of five input variables to the model.

It should be noted that by going from 2D to 3D, and by adding more input parameters for material properties and temperature cycle, training time will be significantly increased. Simultaneously implementing multiple extensions may push the computational power required for training beyond the limits of most individual computers.

It should be noted that as shown in this paper, incorporating theory and knowledge of the physical problem is quite advantageous in machine learning. Although the same PINN architecture proposed here would not be appropriate for other PDEs with different underlying physics, our method potentially can be extended to other physical problems by incorporating related domain knowledge. The generalization of the method, however, was not explored in the current study.

6. Conclusions

In this study, a physics-informed machine learning approach was developed to solve the heat transfer PDE with convective BCs. This was based on training a neural network using a total loss function defined to simultaneously satisfies the PDE, BCs and IC. In addition, physics-informed features were defined based on the heat transfer theory. For accurate training, an adaptive normalizing scheme was developed and implemented to address the difference in magnitudes of loss terms. In addition, the selection of training points in each batch was designed to increase the density of training points around the discontinuities of loss function and inputs.

The predictions of the trained PINN were validated in several 1D and 2D heat transfer cases by comparison with FE results. In addition, performance and accuracy of the PINN was compared with those of a NN without feature engineering. It was shown that while both NN and PINN match the FE results within the training zone, only the PINN with engineered features can capture the physics of the problem to make accurate predictions beyond the training zone.

Developing PINNs to solve heat transfer PDE and other similar PDEs offers tremendous advantages over traditional approaches using FE simulations. Once trained, a PINN can be used for near real-time simulation capability of problems with any given BC. For manufacturing problems, considering uncertainties in the process including unknown BCs, such an approach allows for quick evaluation of the problem during processing to develop feedback loops, realizing the Industry 4.0 concept of active manufacturing control based on sensor data.

CRediT authorship contribution statement

Navid Zobeiry: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing - review & editing, Supervision. **Keith D. Humfeld:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing.

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.

References

Agarap, A.F., 2018. Deep learning using Rectified Linear Units (ReLU). arXiv.

Agrawal, A., Choudhary, A., 2016. Perspective: Materials informatics and big data: Realization of the fourth paradigm of science in materials science. APL Mater. 4, http://dx.doi.org/10.1063/1.4946894.

Barzegar Gerdroodbary, M., 2020. Application of neural network on heat transfer enhancement of magnetohydrodynamic nanofluid. Heat Transfer Res. 49, 197–212. http://dx.doi.org/10.1002/htj.21606.

Erol, S., Schuhmacher, A., Sihn, W., 2016. Strategic guidance towards industry 4.0 - a three-stage process model. In: International Conference on Competitive Manufacturing (COMA). Stellenbosch, South Africa. pp. 495–500.

Fabris, J., Lussier, D., Zobeiry, N., Mobuchon, C., Poursartip, A., 2014. Development of standardized approaches to thermal management in composites manufacturing. In: International SAMPE Technical Conference.

Fabris, J., Mobuchon, C., Zobeiry, N., Lussier, D., Fernlund, G., Poursartip, A., 2015. Introducing thermal history producibility assessment at conceptual design. In: International SAMPE Technical Conference.

Fabris, J., Mobuchon, C., Zobeiry, N., Poursartip, A., 2016. Understanding the consequences of tooling design choices on thermal history in composites processing. In: International SAMPE Technical Conference.

Fernlund, G., Mobuchon, C., Zobeiry, N., 2018. 2.3 Autoclave Processing. In: Carl Zweben, P.B. (Ed.), Comprehensive Composite Materials II. pp. 42–62. http://dx.doi.org/10.1016/b978-0-12-803581-8.09899-4.

Han, J., Jentzen, A., Weinan, E., 2018. Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. USA 115, 8505–8510. http://dx.doi.org/10.1073/pnas.1718942115.

Incropera, F.P., DeWitt, D.P., Bergman, T.L., Lavine, A.S., 2007. Fundamentals of Heat and Mass Transfer, sixth ed. http://dx.doi.org/10.1016/j.applthermaleng.2011.03.022, Fundamentals of Heat and Mass Transfer 6th Edition.

Jagtap, A.D., Karniadakis, G.E., 2019. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J. Comput. Phys. 404, http://dx.doi.org/10.1016/j.jcp.2019.109136.

Johnston, A., Hubert, P., Fernlund, G., Vaziri, R., Poursartip, A., 1996. Process modelling of composite structures employing a virtual autoclave concept. Sci. Eng. Compos. Mater..

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2017. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. 29, 2318–2331. http://dx.doi.org/10.1109/TKDE.2017.2720168.

Liang, L., Liu, M., Martin, C., Sun, W., 2018. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. J. R. Soc. Interface http://dx.doi.org/10.1098/rsif.2017.0844.

Lu, L., Meng, X., Mao, Z., Karniadakis, G.E., 2019. Deepxde: A deep learning library for solving differential equations. arXiv.

Manohar, K., Hogan, T., Buttrick, J., Banerjee, A.G., Kutz, J.N., Brunton, S.L., 2018. Predicting shim gaps in aircraft assembly with machine learning and sparse sensing. J. Manuf. Syst. 48, 87–95. http://dx.doi.org/10.1016/j.jmsy.2018.01.011.

Mohseni, M., Zobeiry, N., Fernlund, G., 2019. Experimental and numerical study of coupled gas and resin transport and its effect on porosity. J. Reinf. Plast. Compos. 38, 1055–1066. http://dx.doi.org/10.1177/0731684419865783.

Mosavat, M., Moradi, R., Rahimi Takami, M., Barzegar Gerdroodbary, M., Ganji, D.D., 2018. Heat transfer study of mechanical face seal and fin by analytical method. Eng. Sci. Technol. Int. J. 21, 380–388. http://dx.doi.org/10.1016/j.jestch.2018.05.001.

Park, J., Zobeiry, N., Poursartip, A., 2017. Tooling materials and their effect on surface thermal gradients. In: International SAMPE Technical Conference.

Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: 30th International Conference on Machine Learning, ICML 2013.

Raissi, M., 2018. Deep hidden physics models: Deep learning of nonlinear partial differential equations. J. Mach. Learn. Res..

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017a. Physics Informed Deep Learning (Part II): Data-Driven Discovery of Nonlinear Partial Differential Equations.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017b. Physics Informed Deep Learning (Part I): Data-Driven Solutions of Nonlinear Partial Differential Equations, Vol. 1. pp. 17–19.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. http://dx.doi.org/10.1016/j.jcp.2018.10.045.

Sheikholeslami, M., Gerdroodbary, M.B., Moradi, R., Shafee, A., Li, Z., 2019. Numerical mesoscopic method for transportation of H2O-based nanofluid through a porous channel considering Lorentz forces. Internat. J. Modern Phys. C 30, http://dx.doi.org/10.1142/S0129183119500074.

Singh, A.P., Medida, S., Duraisamy, K., 2017. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. AIAA J. 55, 2215–2227. http://dx.doi.org/10.2514/1.J055595.

Sirignano, J., Spiliopoulos, K., 2018. DGM: A deep learning algorithm for solving partial differential equations. J. Comput. Phys. 375, 1339–1364. http://dx.doi.org/10.1016/j.jcp.2018.08.029.

Sun, L., Gao, H., Pan, S., Wang, J.X., 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput. Methods Appl. Mech. Engrg. 361, 112732. http://dx.doi.org/10.1016/j.cma.2019.112732.

Tlili, I., Moradi, R., Barzegar Gerdroodbary, M., 2019. Transient nanofluid squeezing cooling process using aluminum oxide nanoparticle. Internat. J. Modern Phys. C 30, http://dx.doi.org/10.1142/S0129183119500785.

Wagner, N., Rondinelli, J.M., 2016. Theory-guided machine learning in materials science. Front. Mater. 3, 28. http://dx.doi.org/10.3389/fmats.2016.00028.

Wang, S., Teng, Y., Perdikaris, P., 2020. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv.

Wang, J.X., Wu, J.L., Xiao, H., 2017. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys. Rev. Fluids 2, 034603. http://dx.doi.org/10.1103/PhysRevFluids.2.034603.

Zobeiry, N., Duffner, C., 2018. Measuring the negative pressure during processing of advanced composites. Compos. Struct. 203, 11–17. http://dx.doi.org/10.1016/j.compstruct.2018.06.123.

Zobeiry, N., Forghani, A., Li, C., Gordnian, K., Thorpe, R., Vaziri, R., Fernlund, G., Poursartip, A., 2016. Multiscale characterization and representation of composite materials during processing. Phil. Trans. R. Soc. A 20150278. http://dx.doi.org/10.1098/rsta.2015.0278.

Zobeiry, N., Humfeld, K.D.K.D., 2019. An iterative scientific machine learning approach for Discovery of Theories Underlying Physical Phenomena. arXiv Prepr. arXiv:1909.

Zobeiry, N., Lee, A., Mobuchon, C., 2020a. Fabrication of transparent advanced composites. Compos. Sci. Technol. 197, 108281. http://dx.doi.org/10.1016/j.compscitech.2020.108281.

Zobeiry, N., Park, J., Poursartip, A., 2019a. An infrared thermography-based method for the evaluation of the thermal response of tooling for composites manufacturing. J. Compos. Mater. 53, 1277–1290. http://dx.doi.org/10.1177/0021998318798444.

Zobeiry, N., Poursartip, A., 2015. The origins of residual stress and its evaluation in composite materials. In: Structural Integrity and Durability of Advanced Composites: Innovative Modelling Methods and Intelligent Design. Elsevier Ltd, pp. 43–72. http://dx.doi.org/10.1016/B978-0-08-100137-0.00003-1.

Zobeiry, N., Reiner, J., Vaziri, R., 2020b. Theory-guided machine learning for damage characterization of composites. Compos. Struct. 246, 112407. http://dx.doi.org/10.1016/j.compstruct.2020.112407.

Zobeiry, N., Stewart, A., Poursartip, A., 2020c. Applications of machine learning for process modeling of composites. In: SAMPE Virtual Conference.

Zobeiry, N., VanEe, D., Anthony, F., Poursartip, A., 2019b. Theory-guided machine learning for process simulation of composites theory-guided machine learning composites processing modelling for manufacturability assessment in preliminary design. In: In NAFEMS 17th World Congress. Quebec City, Canada.