

Physics-informed neural networks for estimating stress transfer mechanics in single lap joints^{#*}

Shivam SHARMA¹, Rajneesh AWASTHI¹, Yedlabala Sudhir SASTRY², Pattabhi Ramaiah BUDARAPU^{†‡1}

¹*School of Mechanical Sciences, Indian Institute of Technology, Bhubaneswar 752050, India*

²*Department of Aeronautical Engineering, Institute of Aeronautical Engineering, Hyderabad 500043, India*

[†]E-mail: pattabhi@iitbbs.ac.in

Received Sept. 8, 2020; Revision accepted Feb. 24, 2021; Crosschecked July 20, 2021

Abstract: With the explosive growth of computational resources and data generation, deep machine learning has been successfully employed in various applications. One important and emerging scientific application of deep learning involves solving differential equations. Here, physics-informed neural networks (PINNs) are developed to solve the differential equations associated with a specific scientific problem. As such, algorithms for solving the differential equations by embedding their initial and boundary conditions in the cost function of the artificial neural networks using algorithmic differentiation must also be developed. In this study, various PINNs are adopted to estimate the stresses in the tablets and the interphase of a single lap joint. The proposed model is represented by two fourth-order non-homogeneous coupled partial differential equations, with the axial stresses in the upper and lower tablets adopted as the dependent variables. The axial stresses are a function of the tablet length, which presents the independent variable. Therefore, the axial stresses in the tablets are estimated by solving the coupled partial differential equations when subjected to the boundary conditions, whereas the remaining stress components are expressed in terms of axial stresses. The results obtained using the developed methodology are validated using the results obtained via MAPLE software.

Key words: Physics-informed neural networks (PINNs); Algorithmic differentiation; Artificial neural networks; Loss function; Single lap joint

<https://doi.org/10.1631/jzus.A2000403>

CLC number: TP183

1 Introduction


The working mechanics of neural networks can be correlated with those of the human brain, which comprises billions of neurons. The concept behind artificial neural networks (ANNs) is to reduce a given task to the minimization of an objective function.

The main advantage of ANNs is their capacity to efficiently perform parallel mathematical operations using high-level open source libraries such as TensorFlow, PyTorch, and Keras. This technique has been extended for solving numerous ordinary and partial differential equations (PDEs) (Yadav et al., 2015). In fact, the use of ANNs in conjunction with classical methods can lead to the development of efficient solution techniques for various physical problems involving PDEs. Meanwhile, physics-informed neural networks (PINNs) are neural networks trained to solve supervised learning tasks while satisfying the applicable laws of physics described by general nonlinear PDEs (Fang and Zhan, 2020). These PINNs can replace the traditional discretization methods with

[‡] Corresponding author

* Project supported by the Science and Engineering Research Board (SERB), Department of Science and Technology (DST), India (No. SRG/2019/001581)

[#] Electronic supplementary materials: The online version of this article (<https://doi.org/10.1631/jzus.A2000403>) contains supplementary materials, which are available to authorized users

 ORCID: Shivam SHARMA, <https://orcid.org/0000-0003-4148-1624>; Pattabhi Ramaiah BUDARAPU, <https://orcid.org/0000-0001-9884-1622>

a neural network that approximates the solutions of differential equations (Raissi et al., 2019; Kadeethum et al., 2020). A PINN algorithm for solving brittle fracture problems by minimizing the variational energy of the system is presented by Goswami et al. (2020), where the boundary conditions are entirely satisfied via appropriate modification of the neural network output.

In recent years, deep machine learning (DML) has been successfully applied in various fields, including academia (Waheed et al., 2020), financial markets (Yang et al., 2020), and nephrology (Thongprayoon et al., 2020). This is mainly due to the exponential growth of computational resources and data generation following significant scientific developments (Chen et al., 2014; Gupta et al., 2020). Samaniego et al. (2020) explored the area of deep neural networks for solving various PDEs in science and engineering, specifically in terms of mechanical problems based on the energy and natural loss function for a machine learning method. Meanwhile, a combined ANN/adaptive-collocation-strategy-based method for solving PDEs was introduced by Anitescu et al. (2019), where variable grid points are employed based on the value of the residual. Elsewhere, a deep collocation method using a feed-forward deep neural network to solve thin plate bending problems was proposed by Guo et al. (2019), and the applications of machine learning in composite materials modeling and design were reviewed by Chen and Gu (2019). An overview of machine learning applications in composite manufacturing was presented by Sacco et al. (2020), and a machine learning approach for predicting the effective elastic properties of composites with arbitrary shapes and inclusion distributions was proposed by Ye et al. (2019). Finally, a convolutional neural network model for predicting the mechanical properties of a 2D checkerboard composite with both soft and ductile and stiff and brittle phases was discussed by Abueidda et al. (2019).

Nacre is found in the inner layers of mollusk shells, enclosing the mollusk's internal organs to protect the soft and delicate visceral mass (Budarapu et al., 2019). It consists of 95% brittle aragonite (calcium carbonate), with the remainder being organic polysaccharide arranged in form of brick and mortar structure (Barthelat et al., 2007). Such a combination of hard inclusions in a ductile matrix can sustain large deformations (Du et al., 2019; Ma-

grini et al., 2019). The superior mechanical properties of nacre-like structures are because of the presence of a weak interface, which prevents the brittle failure of tablets under tension. In addition, small-sized tablets confer stronger and increased tensile strength compared with bulk aragonite. The underlying nano-mechanics governing the structural resilience and absorption of the mechanical energy of nacre is attributed to the nanoscale recovery of heavy deformation (Gim et al., 2019). Pan et al. (2017) proposed a thermally conductive polymer composite based on a bio-inspired naturally layered nacre-like brick and mortar structure. Moreover, Morsali et al. (2020) developed a combined statistical analysis and machine-learning-based computational approach to estimate the structure-property design strategies for nacre-like brick and mortar composites. The machine learning techniques were combined with specific tomography data to analyze the nacreous architecture in (Beliaev et al., 2020). The staggered arrangement of brick and mortar in nacre results in overlapping tablets connected by a polymer matrix, which can be further extended to develop optimized nacre-like materials with an optimum composition, microstructure, and size (Bertoldi et al., 2008; Barthelat, 2014).

Lap joint configurations involve a multilayer structure comprising two substrates and an adhesive layer, with dissimilar geometric and material properties of the adherends that are common in various applications (Wang and Rose, 2003). As a result of multiple layers with varying properties, the stress states prevailing at various levels in a bonded joint tend to be complex; therefore, considerable efforts have been devoted for developing accurate yet simple analytical methods for various types of lap joint, including double overlap, single lap, and single strap joints. Several valuable reports are available in the existing literature, including that from Adams and Mallick (1992). However, the accurate determination of the stresses at the critical locations within a joint requires detailed numerical computations using 3D elastic-plastic finite element analysis (Reinoso et al., 2019) or elaborate experimental techniques (Wang and Rose, 2003). The temperature trends during friction-assisted joining as a function of different process parameters were investigated in (Lambiase et al., 2020) via the training, calibration, and validation of an ANN. A neural network and

artificial intelligence (AI)-based model for studying the effects of welding parameters on the bead penetration area and predicting the bead penetration area for lap joints in the welding process was reported in (Oh et al., 2019).

An analytical model for predicting the interfacial shear and peel stresses in a double lap joint was discussed by Her and Chan (2019). Several experimental studies on adhesive-bonded carbon-fiber-reinforced plastics (Budarapu et al., 2019; Gupta et al., 2019), joggled lap joints, and single lap joints are conducted in (Barile et al., 2020), with their attendant failure strengths reported. Rangaswamy et al. (2020) experimentally investigated the influence of overlap length, adhesive thickness, and adherend surface preparations on the failure behavior of adhesive-bonded joints using trained neural networks, whereas the failure load in single lap adhesive joints subjected to tensile loading was estimated using ANNs by Tosun and Çalık (2016). Gunes et al. (2011) employed ANNs and various genetic algorithms to study the influence of design parameters of the adhesive joint in their free vibration analysis (Budarapu et al., 2014) of an adhesively bonded functionally graded tubular single lap joint. Furthermore, the hygrothermal aging effect on the damage behavior of adhesive composite single lap joints was studied (Xu et al., 2019) using the acoustic emission technique in (Budarapu et al., 2015).

In this study, both the interfacial and normal stresses in a lap joint are estimated using a number of PINNs. A mathematical model for overlapping tablets joined by an adherend, which is represented by two fourth-order non-homogeneous coupled PDEs, is proposed, with the axial stresses in the upper and lower tablets adopted as the variables. As such, the axial stresses in tablets are estimated by solving the coupled PDEs when subjected to the boundary conditions, whereas the axial stress in the interphase is estimated by considering the force equilibrium. The remaining stress components, i.e. stresses along the y direction and shear stresses, are expressed in terms of axial stresses. The estimated stresses can be further used to estimate the fracture toughness of lap joints. The major features of the present study are as follows:

1. The creation of a novel PINN-based DML approach to solve two non-homogeneous coupled fourth-order PDEs.

2. The validation of the results obtained via the developed methodology using the results obtained via MAPLE software.

3. The inclusion of full Python script to estimate all the stress components in both the tablets and interphase.

2 Methodology

Machine learning involves the training of machines to learn from the data prior to its application for making logical decisions; this is achieved using appropriate algorithms. Furthermore, the decision-making capabilities, such as confirming the correctness of the estimated results, can be incorporated via DML, which involves employing the relevant neural networks and algorithms with the trained machines. Therefore, DML consists of some hidden/extra layers (up to 150) on top of the traditional neural networks, which generally contain 2–3 hidden layers. Deep learning is a subset of machine learning in AI that involves networks capable of unsupervised learning from unstructured or unlabeled data. In general, deep learning is employed to perform end-to-end learning, where the network is given raw data and a specific task to perform. A key advantage of deep learning networks is that they generally continue to improve as the size of the data increases.

2.1 Physics-informed neural networks

A typical ANN operates like the biological (human) neural networks. Feed-forward neural networks are ANNs in which the connections between the units do not form a cycle. Here, the information flows only in a forward direction in the network without any loops, initially through the input nodes, followed by the hidden nodes, and finally through the output nodes. Feed-forward neural networks are primarily used for supervised learning where the data to be learned is neither sequential nor time-dependent; however, with the help of the universal approximation theorem, feed-forward neural networks can be used to compute a continuous real-value function. In general, an ANN consists of an input layer, generally a number of hidden layers, and an output layer (Fig. 1). The hidden layers do not exist in a single-layer ANN; it presents a prototypical m -layer ANN that computes a 1D output (OP) on an n -dimensional input $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$.

For example, assuming that a feed-forward neural network is employed to compute a function $u(\mathbf{x}; \Theta)$ on a fixed-sized input \mathbf{x} and Θ , such that the network is set up by \hat{u} with training pairs $(\mathbf{x}; \Theta)$. Here, the derivatives of the network $u(\mathbf{x}; \Theta)$ with respect to its input \mathbf{x} are estimated by applying the chain rule using algorithmic differentiation (AD), which is integrated in deep learning packages such as TensorFlow and PyTorch. In fact, AD is more efficient than the finite difference method, especially for input variables with a larger size compared with the dimensionality of the output variables. This is because the time complexity for reverse mode AD depends on the dimensionality of the output variables, whereas in the finite difference method, it depends on the dimensionality of the input variables. The training data \mathbf{T} in the problem domain Ω with a boundary indicated by $\partial\Omega$ is divided into two sets, $\mathbf{T}_f \in \Omega$ and $\mathbf{T}_b \in \partial\Omega$, and the loss function is defined as the weighted summation of the L^2 norm of error as follows:

$$L(\Theta; \mathbf{L}) = w_f L_f(\Theta; \mathbf{T}_f) + w_b L_b(\Theta; \mathbf{T}_b), \quad (1)$$

where $\Theta = \{\mathbf{W}_l, \mathbf{b}_l\}$ contains the weight matrices and bias vectors of all the neurons in the neural network u , and w_f and w_b represent the weights of the points in the domain and on the boundary, respectively. Thus, after constructing the neural network and the suitable loss function, the objective is to estimate a good approximation for Θ by minimizing the loss function L . The loss function can be minimized using various gradient optimization techniques, such as the Adam and L-BFGS methods.

However, the included Python script also contains commented commands to perform the optimization using the gradient descent optimizer.

Fig. 1 shows a schematic of the PINN to solve the PDEs while considering the boundary and initial conditions, where the feed-forward neural network is highlighted in the neural network block. As the figure shows, an estimate of the stresses σ_1 and σ_3 was performed using the dimensions and material properties of the constituents of a lap joint. Furthermore, the derivatives of the stresses σ_1 and σ_3 were estimated via AD, which were subsequently used for the calculation of the loss function. The loss function was minimized through an iterative process to obtain the final values of the stresses σ_1 and σ_3 such that the boundary conditions were satisfied.

3 Analysis of single lap joint

3.1 Analytical model of single lap joint

Considering a single lap joint, as shown in Fig. 2, where two tablets with a length L and thicknesses h_1 and h_2 are bonded by a matrix with a length L and thickness h . Here, the origin of the coordinate system is placed at the lower-left corner, and the distances of the top of the bottom tablet, matrix, and top tablet from the bottom surface are denoted by b , c , and d , respectively. When the right edge of the top tablet is subjected to a uniaxial tensile load σ_0 , the reaction force on the left edge of the bottom tablet can be given by $\rho\sigma_0$, where $\rho = h_1/h_2$.

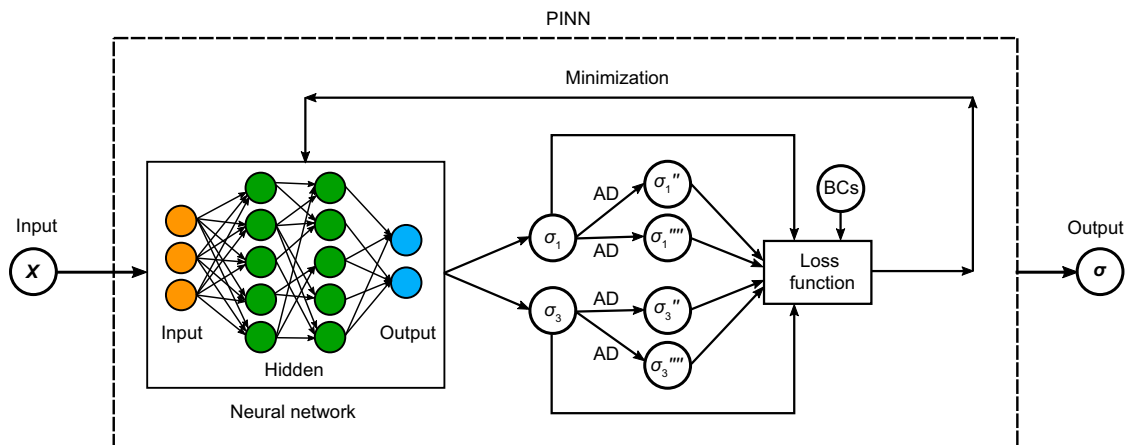


Fig. 1 Schematic of the PINN to solve the PDEs while considering the boundary conditions (BCs) and initial conditions

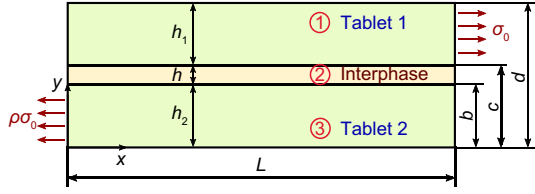


Fig. 2 Schematic of two tablets with a length L and thicknesses h_1 and h_2 bonded by an interphase (matrix) with a length L and thickness h in the middle. An axial load σ_0 was imposed on the right edge of the top tablet parallel to the tablet length, whereas the bottom tablet was fixed. The geometry was based on a nacre-like brick and mortar structure

Let the normal stresses along the x and y directions be denoted by $\sigma_x^{(i)}$ and $\sigma_y^{(i)}$, respectively, and the shear stresses by $\tau_{xy}^{(i)}$, where the superscript i indicates the constituent $i = 1, 2,$ and 3 , which represent the top tablet, matrix, and bottom tablet, respectively (Fig. 2). Here the equilibrium equations along the x and y directions can be written as follows (Ajayan et al., 2003):

$$\frac{\partial \sigma_x^{(i)}}{\partial x} + \frac{\partial \tau_{xy}^{(i)}}{\partial y} = 0, \tag{2a}$$

$$\frac{\partial \tau_{xy}^{(i)}}{\partial x} + \frac{\partial \sigma_y^{(i)}}{\partial y} = 0, \tag{2b}$$

where the boundary conditions are given by:

$$\text{At } x = 0: \sigma_x^{(3)} = \rho\sigma_0, \tau_{xy}^{(3)} = 0, \tag{3a}$$

$$\sigma_x^{(1)} = 0, \tau_{xy}^{(1)} = 0;$$

$$\text{At } x = L: \sigma_x^{(1)} = \sigma_0, \tau_{xy}^{(1)} = 0, \tag{3b}$$

$$\sigma_x^{(3)} = 0, \tau_{xy}^{(3)} = 0;$$

$$\text{At } y = c: \tau_{xy}^{(2)} = \tau_{xy}^{(1)}. \tag{3c}$$

Combining the governing equations (Eq. (2)) and the boundary conditions described in Eq. (3) after denoting $\sigma_x^{(1)} = \sigma_1$, $\sigma_x^{(2)} = \sigma_2$, and $\sigma_x^{(3)} = \sigma_3$, the stress components in the individual constituents can be estimated as follows:

$$\sigma_y^{(1)} = \frac{1}{2}(d - y)^2 \sigma_1'', \tag{4a}$$

$$\tau_{xy}^{(1)} = (d - y) \sigma_1', \tag{4b}$$

$$\sigma_y^{(2)} = \frac{1}{2}(c - y)^2 \sigma_2'' + h_1 \left(c - y + \frac{h_1}{2} \right) \sigma_1'', \tag{4c}$$

$$\tau_{xy}^{(2)} = (c - y) \sigma_2' + h_1 \sigma_1', \tag{4d}$$

$$\sigma_y^{(3)} = \frac{1}{2}[(y^2 - b^2) \sigma_3'' + h^2 \sigma_2'' + (2h_1 h + h_1^2) \sigma_1''], \tag{4e}$$

$$\tau_{xy}^{(3)} = -\sigma_3' y. \tag{4f}$$

Furthermore, considering the force equilibrium, σ_2 can be expressed in terms of σ_1 and σ_3 as follows:

$$\sigma_1 h_1 + \sigma_2 h + \sigma_3 h_2 = \sigma_0 h_1, \tag{5a}$$

$$\sigma_2 = \alpha_1(\sigma_0 - \sigma_1) - \alpha_2 \sigma_3, \tag{5b}$$

where $\alpha_1 = h_1/h$ and $\alpha_2 = h_2/h$. The axial stresses σ_1 and σ_3 were estimated by minimizing the system potential energy, which can be expressed as

$$\Pi^* = \int_0^L \int_y \left(\frac{1}{2} \sigma_x^{(i)} \epsilon_x^{(i)} + \frac{1}{2} \sigma_y^{(i)} \epsilon_y^{(i)} + \tau_{xy}^{(i)} \epsilon_{xy}^{(i)} \right) dx dy, \tag{6}$$

where ϵ_x and ϵ_y indicate the normal strains along the x and y directions, respectively, and ϵ_{xy} represents the shear strain in the xy plane. The properties of the tablet and interphase used in this study are presented in Table 1.

Therefore, considering the stress components in each of the constituents and the material properties presented in Table 1, the potential energy of the system can be estimated using Eq. (6). The estimated system potential energy was further minimized using the Euler-Lagrange variation principles, which resulted in the following simultaneous equations related to the axial stresses σ_1 and σ_3 in tablets 1 and 2, respectively:

$$5.4968 \times 10^{-9} \sigma_1'''' + 1.3410 \times 10^{-9} \sigma_3'''' \tag{7a}$$

$$+ 1.4805 \times 10^{-6} \sigma_1'' + 1.4400 \times 10^{-6} \sigma_3'' \\ + 3.8407 \times 10^{-4} \sigma_1 + 3.8400 \times 10^{-4} \sigma_3 \\ - 3.8400 \times 10^{-4} = 0,$$

$$1.3410 \times 10^{-9} \sigma_1'''' + 2.0119 \times 10^{-9} \sigma_3'''' \tag{7b}$$

$$+ 1.4400 \times 10^{-6} \sigma_1'' + 5.7390 \times 10^{-7} \sigma_3'' \\ + 3.8400 \times 10^{-4} \sigma_1 + 3.8407 \times 10^{-4} \sigma_3 \\ - 3.8400 \times 10^{-4} = 0,$$

where σ_1 and σ_3 are the axial stresses in tablets 1 and 2, respectively, and the length of the tablets (L) is considered to be 8 μm . The next objective was to solve the coupled partial differential equations (Eq. (7)) using neural networks for estimating the stresses $\sigma_1(x)$ and $\sigma_3(x)$, where x is the length of the tablet defined in the domain $0 \leq x \leq L$. The main steps in the solution algorithm of the developed PINN are summarized below.

1. Construct a neural network with two output variables $\sigma_1, \sigma_3 = \text{NN}(\mathbf{x}; \Theta)$, where Θ indicates the parameters of the neural network, i.e. $\Theta = \{ \mathbf{W}_l, \mathbf{b}_l \}$

Table 1 Material properties of the tablets and the interphase of the single lap joint (Fig. 2) used in this study (Jackson et al., 1988; Barthelat and Espinosa, 2007; Espinosa et al., 2009; Ni et al., 2015)

| Constituent | Young's modulus (GPa) | Poisson's ratio | Shear modulus (GPa) | Length (μm) | Thickness (μm) |
|-------------|-----------------------|-----------------|---------------------|--------------------------|-----------------------------|
| Tablet | 100 | 0.3 | 38.46 | 8 | 0.4 |
| Interphase | 2.8 | 0.4 | 1.0 | 8 | 0.025 |

is the set of all the weight matrices and bias vectors of the neural network, where $1 \leq l \leq n_l$ and n_l denotes the number of layers.

2. Prepare a training set N_f for training the above constructed neural network. Each point in the training set lies within the specified domain of the differential equations.

3. Estimate the loss function by calculating the average L^2 norm of the differential equations and the boundary conditions over the whole training set. The loss function is given by the following:

$$L^{(i)} = w_f L_f^{(i)} + w_b L_b^{(i)}, \quad (8a)$$

$$\text{Loss} = \frac{1}{N_f} \sum_{i=0}^{N_f} L^{(i)}, \quad (8b)$$

where

$$L_f = |f(\sigma_1, \sigma_3)|^2 + |g(\sigma_1, \sigma_3)|^2, \quad (9a)$$

$$L_b = |\sigma_1(x=0)|^2 + |\sigma_3(x=0) - \rho\sigma_0|^2 \quad (9b)$$

$$+ |\sigma_1(x=L) - \sigma_0|^2 + |\sigma_3(x=L)|^2$$

$$+ |\sigma_1'(x=0)|^2 + |\sigma_3'(x=0)|^2 + |\sigma_1'(x=L)|^2$$

$$+ |\sigma_3'(x=L)|^2,$$

and functions $f(\sigma_1, \sigma_3)$ and $g(\sigma_1, \sigma_3)$ are given by

$$f(\sigma_1, \sigma_3) = 5.4968 \times 10^{-9} \sigma_1'''' \quad (10a)$$

$$+ 1.3410 \times 10^{-9} \sigma_3'''' + 1.4805 \times 10^{-6} \sigma_1''$$

$$+ 1.4400 \times 10^{-6} \sigma_3'' + 3.8407 \times 10^{-4} \sigma_1$$

$$+ 3.8400 \times 10^{-4} \sigma_3 - 3.8400 \times 10^{-4} = 0,$$

$$g(\sigma_1, \sigma_3) = 1.3410 \times 10^{-9} \sigma_1'''' \quad (10b)$$

$$+ 2.0119 \times 10^{-9} \sigma_3'''' + 1.4400 \times 10^{-6} \sigma_1''$$

$$+ 5.7390 \times 10^{-7} \sigma_3'' + 3.8400 \times 10^{-4} \sigma_1$$

$$+ 3.8407 \times 10^{-4} \sigma_3 - 3.8400 \times 10^{-4} = 0.$$

The derivative of the output variables σ_1 and σ_3 of the neural network with respect to the input \mathbf{x} was calculated by applying the chain rule of differentiation using AD, an inbuilt function of the TensorFlow deep learning framework. The loss function was prepared in such a way that the output of the neural

network satisfied the differential equations as well as the imposed boundary conditions for all the points in the training set.

4. Optimize the loss function in Eq. (8) using the Adam and L-BFGS gradient-based optimization solvers to determine the optimal parameter Θ^* for the proposed neural network.

3.2 Implementation aspects of the neural network schema

The neural network was implemented using Python language version 3.6.9 in TensorFlow framework version 1.15.2; the developed neural network was trained using a Tesla T4 GPU card in Google Colab. An iterative process was followed in this study to ensure that the architecture of the neural network reached the least value of the loss function. A fully connected feed-forward artificial neural network architecture with three hidden layers (each layer contained 500 neurons), one input layer, and one output layer containing two output neurons was selected for the analysis. The network was comprehensively tested with varying number of neurons in each layer to test the convergence and repeatability of the results; the developed network produced stable results every time. The histograms formed during the network training are presented in Section S2 of the electronic supplementary materials (ESM), and the network architecture in the Python script is described in Listing S1 in the ESM. The implementation of the developed PINN is described in Listing S2 in the ESM.

The training data was obtained using Latin hypercube sampling (LHS), where the size of the training data set generated (i.e. N_f [training data]) was 5000. One-dimensional LHS involves dividing the cumulative density function into n equal partitions and then choosing a random data point in each partition. This type of sampling significantly reduces the number of iterations required to achieve a reasonable and accurate result. In fact, well-performed LHS can reduce the processing time by up to 50%;

on the contrary, the LHS can lead to more accurate results when there is no time constraint. The training data sets in the Python script are described in Listing S3 in the ESM.

The initialization of the network using the correct weights can result in quick convergence with stable network loss function for a large number of iterations (hundreds of thousands). Therefore, the initialization step is critical to the model's ultimate performance. In this study, we used the Xavier initialization function for the initialization of the neural network, i.e. initialization of the weights and biases. Xavier initialization involves controlling the constant variance across every layer, with the output centered around 0, indicating that the mean is 0. The variance of weights in layer l is given by the following equation:

$$\text{var} \left(W^{(i)} \right) = \frac{1}{n^{(l-1)}}, \quad (11)$$

where $n^{(l-1)}$ represents the number of neurons in the previous layer. The initialization steps in the Python script are presented in Listing S4 in the ESM. The initial parameters were set based on previous experience, which will be fine-tuned later.

Meanwhile, activation functions were incorporated in the neural network to introduce nonlinearity. These functions are generally employed to convert an input signal of a node in an ANN to an output signal, and the converted output signal is further used as an input to the next layer in the stack. In this study, the hyperbolic tangent activation function (Tanh) was employed as the activation function. However, the included Python script also contains commented commands to estimate the activation function using Sigmoid and rectified linear unit (ReLU) functions. The neural network parameters were optimized based on the loss function. As such, the objective was to minimize the loss for a given neural network by optimizing its parameters (weights and biases). In this study, we adopted a mean square error loss function, which is the square of the difference between the predicted value and that obtained via the neural network considering the boundary conditions and the predicted output based on the training values. The estimation of the loss function using the Python script is described in Listing S5 in the ESM.

Optimization algorithms were subsequently adopted to optimize the objective function, i.e. the

loss function. Here, the mean square error loss function, as described above, was employed. When the loss function was close to zero, the solution could be deemed to be converging. The computer implementation program, developed in TensorFlow using Python script, for estimating the loss function and thus the stresses, is attached in the ESM.

3.3 Results and discussion

The series of differential equations developed in Section 3.1 was solved using the neural network model proposed in Section 3.2. Two different types of activation functions, the hyperbolic tangent (Tanh) and ReLU functions, were tested using the neural network architecture, with the results subsequently compared. Here, the results obtained via the ReLU activation function clearly required less computational time for each epoch compared with the Tanh activation function. This was evident given the number of epochs trained in the same time period for both the activation functions. However, the results estimated using the Tanh activation function were found to be more reliable and stable and were thus adopted for the subsequent analysis. This was because the Tanh activation function provided both positive and negative values, thus yielding more accurate results.

The variation in loss function while training the proposed PINN as a function of the number of iterations is plotted in Fig. 3. As the figure shows, initially, the variation was slightly irregular before it smoothly decreased to approach zero according to the number of training epochs of the model. A combination of L-BFGS-B and Adam optimizers was employed here, with the specified iterations equal to 200. However, the error tolerance for the loss function was set to machine accuracy, which is of the order of 10^{-12} . Therefore, beyond 200 iterations, the system would switch to the Adam optimizer until the

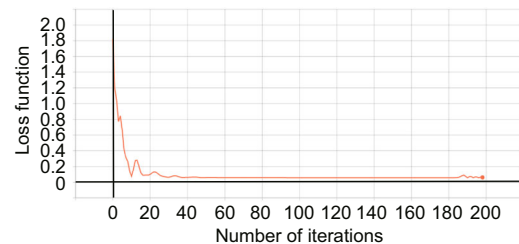


Fig. 3 Variation in loss function during the training of the PINN with a number of iterations

specified error tolerance was achieved. However, the maximum number of iterations under any conditions should not exceed 50 000.

A uniaxial load σ_0 of 100 MPa was applied on the upper tablet parallel to its length while keeping the lower tablet fixed (Fig. 2). The axial stresses in tablets 1 and 2, indicated by σ_1 and σ_3 , respectively, were estimated by minimizing the loss function (Eq. (8)) while being subjected to the boundary conditions described in Eq. (3). The distribution of the various stresses in tablets 1 and 2 and the interphase were plotted, as shown in Figs. 4–6, where all the stresses were extracted at the mid-layer. Meanwhile, the variations in the axial stresses along the length of tablets 1 and 3 and the interphase were plotted, as shown in Figs. 4a and 4b, respectively. Here, the axial stresses σ_1 and σ_3 clearly complemented each other, thus satisfying the boundary conditions.

The variation in the shear stress in the interphase as a function of the length of the lap joint was plotted, as shown in Fig. 5a. To verify the results

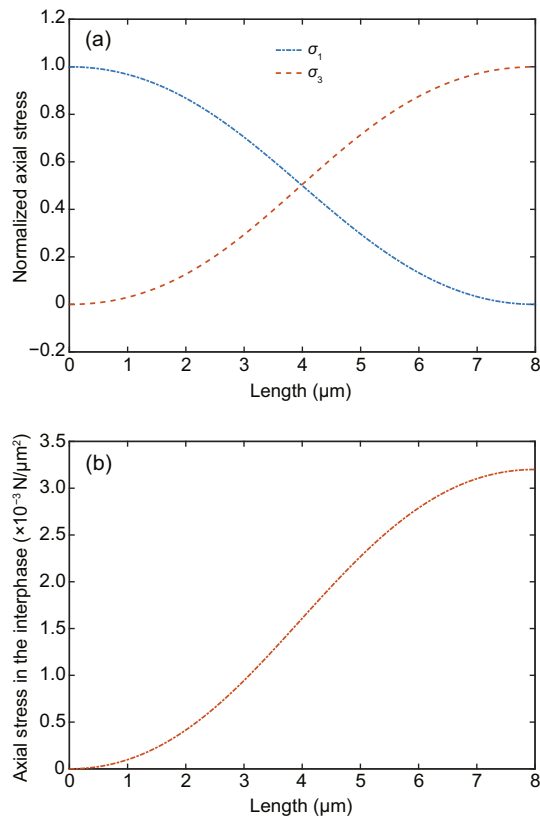


Fig. 4 Distributions of the axial stresses in tablets 1 and 2 (a) and the interphase (b) as a function of tablet length

obtained using the proposed model, the exact solution was estimated by solving the analytical model discussed in Section 3.1 using MAPLE software. A comparison of the interfacial shear stress estimated using the proposed method and the exact solution is shown in Fig. 5a. Here, the exact solution was found to represent the physical phenomena, where the shear stresses at the edges were maximized over a small region and dropped down toward the center. However, as Fig. 5a shows, the results based on the DML approach clearly represented the physical behavior, but they were not in close agreement with the exact solution. Specifically, the proposed DML approach was unable to capture the physical behavior at the interphase edges, which is a prevailing issue with DML-based approaches. This was further confirmed by the comparison of the normal stresses along the y direction in the interphase, as

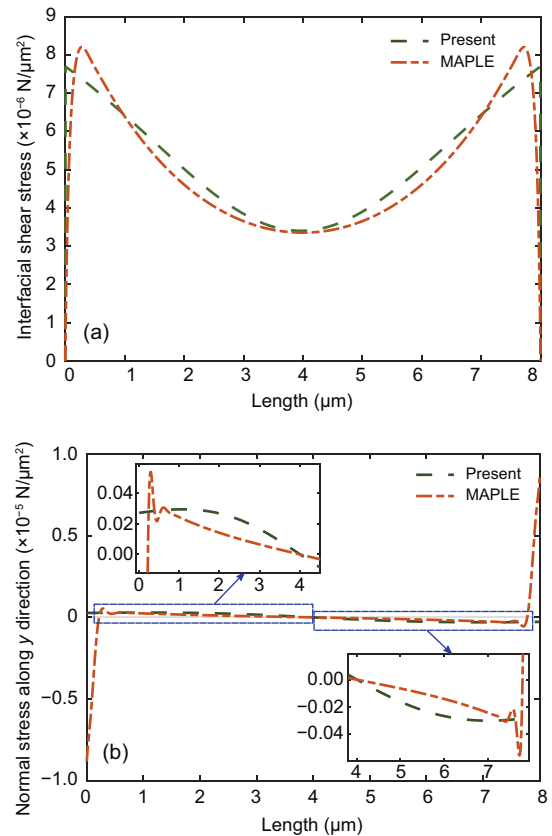


Fig. 5 Distributions of the shear stresses ($\tau_{xy}^{(2)}$) (a) and the normal stresses along the y direction ($\sigma_y^{(2)}$) (b) in the interphase along its length. The results estimated using the developed methodology were compared with those calculated using MAPLE software. Close-ups of the regions on the left and right sides around the center of the interphase are also shown

shown in Fig. 5b, where the solution estimated using the developed DML approach clearly failed at the interphase edges (in the close-up regions). The normalized (with respect to the exact solution) error (difference of DML and exact solutions) in the shear stress estimated at the middle of the joint length was estimated to be 1.22%, and the average normalized error in the normal stress along the y direction in the interphase, calculated as the average of the normalized errors estimated at several points along the interphase length, was 26.19%. The estimated results and errors using the present DML approach are promising; however, advanced mathematical tools can be adopted to minimize the cost function with reduced errors.

The distributions of the normal stresses along the y direction and the shear stresses in tablets 1 and 2 are shown in Figs. 6a and 6b, respectively.

The normal and shear stresses in tablet 2 were significantly lower than those of tablet 1, mainly because the load was applied at the right edge of tablet 1 along the axial direction, i.e. the load was

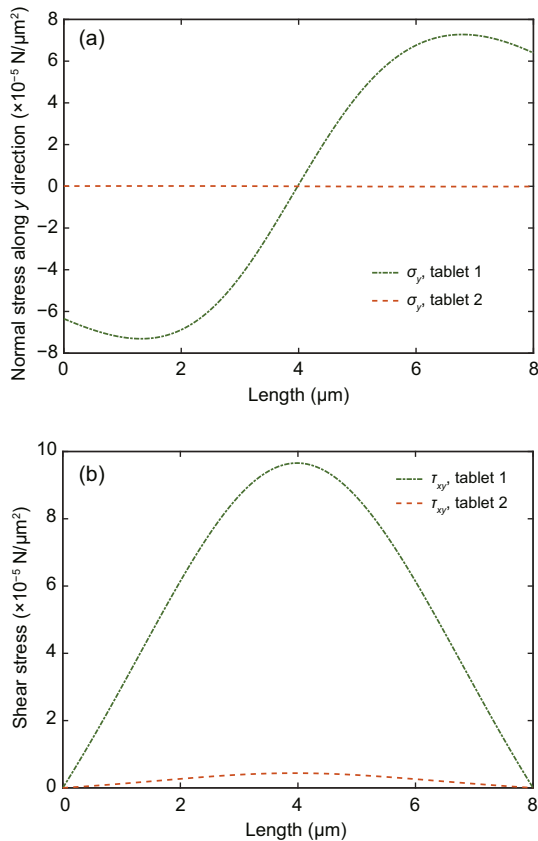


Fig. 6 Distributions of the normal stresses along the y direction (a) and the shear stresses (b) in tablets 1 and 2

shear transferred to tablet 2 through the interphase. Hence, a part of the applied load was dissipated in the interphase in the form of shear and peel stresses.

4 Conclusions

The mechanics of a single lap joint were investigated by solving a DML-based PINN. A uniaxial loading parallel to the tablet length was applied on the right edge of the upper tablet while keeping the lower tablet fixed. The developed framework was then extended to estimate the various stress components in a number of tablets as well as the interphase by solving the coupled fourth-order non-homogeneous PDEs equations subjected to the boundary conditions. The results obtained via the DML-based approach were then compared with the exact solution obtained using MAPLE software.

The interfacial shear stress estimated using the proposed method and the exact solution were also compared. Here, the results obtained via the DML-based approach represented the physical behavior, although they were not in agreement with the exact solution. Specifically, the proposed DML-based approach was unable to capture the physical behavior at the interphase edges. Similar behavior was observed during the comparison of the normal stresses along the y direction in the interphase. The normalized error in the interfacial shear stress estimated at the middle of the joint length was found to be 1.22%, whereas, the average normalized error in the normal stress along the y direction in the interphase was 26.19%.

The method of using deep learning to solve the differential equations is a promising approach because the attendant computer codes are robust enough to handle the changes in the differential equations. Furthermore, the developed deep learning methodology can be used to solve complex equations or a system of equations (e.g. engineered interphases with a tailored elastic modulus). An iterative approach was employed in this paper to determine the correct hyper-parameters, which is essential for obtaining accurate results. Though the training of the neural network requires a computationally expensive approach, we believe that this method will continue to evolve with time, overcoming the above limitations, and it will be used to assist in solving various practical problems in future.

Contributors

Pattabhi Ramaiah BUDARAPU formulated the analytical model and designed the research. Shivam SHARMA developed the DML codes, processed the corresponding data, and wrote the first draft of the manuscript. Rajneesh AWASTHI, Yedlabala Sudhir SASTRY, and Pattabhi Ramaiah BUDARAPU revised and edited the final version.

Conflict of interest

Shivam SHARMA, Rajneesh AWASTHI, Yedlabala Sudhir SASTRY, and Pattabhi Ramaiah BUDARAPU declare that they have no conflict of interest.

References

- Abueidda DW, Almasri M, Ammourah R, et al., 2019. Prediction and optimization of mechanical properties of composites using convolutional neural networks. *Composite Structures*, 227:111264. <https://doi.org/10.1016/j.compstruct.2019.111264>
- Adams RD, Mallick V, 1992. A method for the stress analysis of lap joints. *The Journal of Adhesion*, 38(3-4):199-217. <https://doi.org/10.1080/00218469208030455>
- Ajayan PM, Schadler LS, Braun PV, 2003. Nanocomposite Science and Technology. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany.
- Anitescu C, Atroshchenko E, Alajlan N, et al., 2019. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 59(1):345-359. <https://doi.org/10.32604/cmc.2019.06641>
- Barile C, Casavola C, Moramarco V, et al., 2020. Bonding characteristics of single and joggled-lap CFRP specimens: mechanical and acoustic investigations. *Applied Sciences*, 10(5):1782. <https://doi.org/10.3390/app10051782>
- Barthelat F, 2014. Designing nacre-like materials for simultaneous stiffness, strength and toughness: optimum materials, composition, microstructure and size. *Journal of the Mechanics and Physics of Solids*, 73:22-37. <https://doi.org/10.1016/j.jmps.2014.08.008>
- Barthelat F, Espinosa HD, 2007. An experimental investigation of deformation and fracture of nacre-mother of pearl. *Experimental Mechanics*, 47(3):311-324. <https://doi.org/10.1007/s11340-007-9040-1>
- Barthelat F, Tang H, Zavattieri PD, et al., 2007. On the mechanics of mother-of-pearl: a key feature in the material hierarchical structure. *Journal of the Mechanics and Physics of Solids*, 55(2):306-337. <https://doi.org/10.1016/j.jmps.2006.07.007>
- Beliaev M, Zöllner D, Pacureanu A, et al., 2020. Quantification of sheet nacre morphogenesis using X-ray nanotomography and deep learning. *Journal of Structural Biology*, 209(1):107432. <https://doi.org/10.1016/j.jsb.2019.107432>
- Bertoldi K, Bigoni D, Drugan WJ, 2008. Nacre: an orthotropic and bimodular elastic material. *Composites Science and Technology*, 68(6):1363-1375. <https://doi.org/10.1016/j.compscitech.2007.11.016>
- Budarapu PR, Yb SS, Javvaji B, et al., 2014. Vibration analysis of multi-walled carbon nanotubes embedded in elastic medium. *Frontiers of Structural and Civil Engineering*, 8(2):151-159. <https://doi.org/10.1007/s11709-014-0247-9>
- Budarapu PR, Narayana TSS, Rammohan B, et al., 2015. Directionality of sound radiation from rectangular panels. *Applied Acoustics*, 89:128-140. <https://doi.org/10.1016/j.apacoust.2014.09.006>
- Budarapu PR, Kumar S, Prusty BG, et al., 2019. Stress transfer through the interphase in curved-fiber pullout tests of nanocomposites. *Composites Part B: Engineering*, 165:417-434. <https://doi.org/10.1016/j.compositesb.2018.12.116>
- Chen CT, Gu GX, 2019. Machine learning for composite materials. *MRS Communications*, 9(2):556-566. <https://doi.org/10.1557/mrc.2019.32>
- Chen M, Mao SW, Zhang Y, et al., 2014. Big data generation and acquisition. In: Chen M, Mao SW, Zhang Y, et al. (Eds.), *Big Data: Related Technologies, Challenges and Future Prospects*. Springer, Cham, Switzerland, p.19-32. https://doi.org/10.1007/978-3-319-06245-7_3
- Du GL, Mao AR, Yu JH, et al., 2019. Nacre-mimetic composite with intrinsic self-healing and shape-programming capability. *Nature Communications*, 10(1):800. <https://doi.org/10.1038/s41467-019-08643-x>
- Espinosa HD, Rim JE, Barthelat F, et al., 2009. Merger of structure and material in nacre and bone-perspectives on de novo biomimetic materials. *Progress in Materials Science*, 54(8):1059-1100. <https://doi.org/10.1016/j.pmatsci.2009.05.001>
- Fang ZW, Zhan J, 2020. A physics-informed neural network framework for PDEs on 3D surfaces: time independent problems. *IEEE Access*, 8:26328-26335. <https://doi.org/10.1109/ACCESS.2019.2963390>
- Gim J, Schnitzer N, Otter LM, et al., 2019. Nanoscale deformation mechanics reveal resilience in nacre of *Pinnac nobilis* shell. *Nature Communications*, 10(1):4822. <https://doi.org/10.1038/s41467-019-12743-z>
- Goswami S, Anitescu C, Chakraborty S, et al., 2020. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106:102447. <https://doi.org/10.1016/j.tafmec.2019.102447>
- Gunes R, Apalak MK, Yildirim M, 2011. Free vibration analysis of an adhesively bonded functionally graded tubular single lap joint. *The Journal of Adhesion*, 87(9):902-925. <https://doi.org/10.1080/00218464.2011.600672>
- Guo HW, Zhuang XY, Rabczuk T, 2019. A deep collocation method for the bending analysis of Kirchhoff plate. *Computers, Materials & Continua*, 59(2):433-456. <https://doi.org/10.32604/cmc.2019.06660>
- Gupta S, Modgil S, Gunasekaran A, 2020. Big data in lean six sigma: a review and further research directions. *International Journal of Production Research*, 58(3):947-969. <https://doi.org/10.1080/00207543.2019.1598599>
- Gupta TK, Budarapu PR, Chappidi SR, et al., 2019. Advances in carbon based nanomaterials for bio-medical applications. *Current Medicinal Chemistry*, 26(38):6851-6877. <https://doi.org/10.2174/0929867326666181126113605>

- Her SC, Chan CF, 2019. Interfacial stress analysis of adhesively bonded lap joint. *Materials*, 12(15):2403. <https://doi.org/10.3390/ma12152403>
- Jackson AP, Vincent JFV, Turner RM, 1988. The mechanical design of nacre. *Proceedings of the Royal Society B Biological Sciences*, 234(1277):415-440. <https://doi.org/10.1098/rspb.1988.0056>
- Kadeethum T, Jørgensen TM, Nick HM, 2020. Physics-informed neural networks for solving nonlinear diffusivity and biot's equations. *PLoS One*, 15(5):e0232683. <https://doi.org/10.1371/journal.pone.0232683>
- Lambiase F, Grossi V, Paoletti A, 2020. Machine learning applied for process design of hybrid metal-polymer joints. *Journal of Manufacturing Processes*, 58:92-100. <https://doi.org/10.1016/j.jmapro.2020.08.016>
- Magrini T, Bouville F, Lauria A, et al., 2019. Transparent and tough bulk composites inspired by nacre. *Nature Communications*, 10(1):2794. <https://doi.org/10.1038/s41467-019-10829-2>
- Morsali S, Qian D, Minary-Jolandan M, 2020. Designing bioinspired brick-and-mortar composites using machine learning and statistical learning. *Communications Materials*, 1(1):12. <https://doi.org/10.1038/s43246-020-0012-7>
- Ni Y, Song ZQ, Jiang HY, et al., 2015. Optimization design of strong and tough nacreous nanocomposites through tuning characteristic lengths. *Journal of the Mechanics and Physics of Solids*, 81:41-57. <https://doi.org/10.1016/j.jmps.2015.04.013>
- Oh WB, Yun TJ, Lee BR, et al., 2019. A study on intelligent algorithm to control welding parameters for lap-joint. *Procedia Manufacturing*, 30:48-55. <https://doi.org/10.1016/j.promfg.2019.02.008>
- Pan GR, Yao YM, Zeng XL, et al., 2017. Learning from natural nacre: constructing layered polymer composites with high thermal conductivity. *Applied Materials & Interfaces*, 9(38):33001-33010. <https://doi.org/10.1021/acsami.7b10115>
- Raissi M, Perdikaris P, Karniadakis GE, 2019. 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. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Rangaswamy H, Sogalad I, Basavarajappa S, et al., 2020. Experimental analysis and prediction of strength of adhesive-bonded single-lap composite joints: Taguchi and artificial neural network approaches. *SN Applied Sciences*, 2(6):1055. <https://doi.org/10.1007/s42452-020-2851-8>
- Reinoso J, Durand P, Budarapu PR, et al., 2019. Crack patterns in heterogenous rocks using a combined phase field-cohesive interface modeling approach: a numerical study. *Energies*, 12(6):965. <https://doi.org/10.3390/en12060965>
- Sacco C, Radwan AB, Anderson A, et al., 2020. Machine learning in composites manufacturing: a case study of automated fiber placement inspection. *Composite Structures*, 250:112514. <https://doi.org/10.1016/j.compstruct.2020.112514>
- Samaniego E, Anitescu C, Goswami S, et al., 2020. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362:112790. <https://doi.org/10.1016/j.cma.2019.112790>
- Thongprayoon C, Kaewput W, Kovvuru K, et al., 2020. Promises of big data and artificial intelligence in nephrology and transplantation. *Journal of Clinical Medicine*, 9(4):1107. <https://doi.org/10.3390/jcm9041107>
- Tosun E, Çalık A, 2016. Failure load prediction of single lap adhesive joints using artificial neural networks. *Alexandria Engineering Journal*, 55(2):1341-1346. <https://doi.org/10.1016/j.aej.2016.04.029>
- Waheed H, Hassan SU, Aljohani NR, et al., 2020. Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, 104:106189. <https://doi.org/10.1016/j.chb.2019.106189>
- Wang CH, Rose LRF, 2003. Stress analysis and failure assessment of lap joints. In: Tong LY, Soutis C (Eds.), *Recent Advances in Structural Joints and Repairs for Composite Materials*. Springer, Dordrecht, the Netherlands, p.1-26. https://doi.org/10.1007/978-94-017-0329-1_1
- Xu D, Liu PF, Li JG, et al., 2019. Damage mode identification of adhesive composite joints under hygrothermal environment using acoustic emission and machine learning. *Composite Structures*, 211:351-363. <https://doi.org/10.1016/j.compstruct.2018.12.051>
- Yadav N, Yadav A, Kumar M, 2015. *An Introduction to Neural Network Methods for Differential Equations*. Springer, Dordrecht, the Netherlands. <https://doi.org/10.1007/978-94-017-9816-7>
- Yang RJ, Yu L, Zhao YJ, et al., 2020. Big data analytics for financial Market volatility forecast based on support vector machine. *International Journal of Information Management*, 50:452-462. <https://doi.org/10.1016/j.ijinfomgt.2019.05.027>
- Ye S, Li B, Li QY, et al., 2019. Deep neural network method for predicting the mechanical properties of composites. *Applied Physics Letters*, 115(16):161901. <https://doi.org/10.1063/1.5124529>

List of electronic supplementary materials

- Section S1 Snippets of Python script
- Listing S1 Architecture and initialization of the network
- Listing S2 Implementation of the developed PINN
- Listing S3 Training data
- Listing S4 Xavier initialization steps
- Listing S5 Loss function
- Section S2 Histograms during network training