Journal of the Serbian Society for Computational Mechanics / Vol. 15 / No. 2, 2021 / pp 45-54
(10.24874/jsscm.2021.15.02.05)

PHYSICS-INFORMED NEURAL NETWORKS FOR ELASTIC PLATE PROBLEMS WITH BENDING AND WINKLER-TYPE CONTACT EFFECTS

Aliki D. Muradova, Georgios E. Stavroulakis*

Institute of Computational Mechanics and Optimization, School of Production Engineering and Management, Technical University of Crete, Chania, Greece
e-mail: aliki@mred.tuc.gr; e-mail: gestavr@dpem.tuc.gr
** corresponding author*

Abstract

Kirchhoff plate bending and Winkler-type contact problems with different boundary conditions are solved with the use of physics-informed neural networks (PINN). The PINN is built on the base of mechanics laws and deep learning. The idea of the technique includes fitting the governing partial differential equations at collocation points and then training the neural network with the use of optimization techniques. Training of the neural network is performed by numerical optimization using Adam's method and the L-BFGS (Limited- Broyden–Fletcher–Goldfarb–Shanno) algorithm. The error loss function and the computational error of the approximate solution (output of the neural network) of the bending problem and contact problem with Winkler type elastic foundation are shown on examples. The predictions of the NN are investigated for different values of the foundation's constants. The effectiveness of the proposed framework is demonstrated through numerical experiments with different numbers of epochs, hidden layers, neurons and numbers of collocation points. The Tensorflow deep learning and scientific computing package of Python is used through a Jupyter Notebook.

Keywords: plate bending, Winkler foundation, partial differential equation, physics-informed neural network, deep learning, collocation method.

1. Introduction

In the last years within the framework of data-driven computational techniques a relatively new method, so-called physics-informed neural network (PINN), became widely distributed and popular. The cost of effort and time needed for analyzing complex physical, mechanical, biological and engineering problems can be prohibitive and, in some cases even impossible to construct a numerical solution, converging to an exact solution of the considered problem, based only on classical computational tools. On the contrary, in several applications described by differential equations a quick even less accurate approximation of the solution is required. This task can be accomplished by a PINN.

Artificial neural networks (ANNs) have been developed long time ago (see e.g., McCulloch and Pitts 1943). During recent years thanks to the technological progress in developing of strong, powerful computing software, a big growth of available data and thanks to elaboration of backpropagation techniques and deep learning mechanism, became possible to solve many complicated problems in science and engineering.

For the solution of elastoplastic and contact problems in mechanics by using the minimization of energy, Hopfield and Tank neural networks have been proposed by Kortesis and Panagiotopoulos (1993), and Avdelas et al. (1995). Feedforward NNs trained by the backpropagation algorithm, have been used for the approximation of several problems in mechanics based on examples (supervised learning). Inverse and parameter-identification problems in mechanics have been solved by using backpropagation neural networks by Stavroulakis et al. (1997), Stavroulakis (2000), Waszczyszyn and Ziemiański (2005) and Stavroulakis et al. (2003). Buckling loads in nonlinear problems for elastic plates have been estimated with the use of neural network in the work of Muradova and Stavroulakis (2007). A recent review of classical usage of neural networks within computational mechanics can be found in Yagawa and Oishi (2021).

The technique of using the governing partial differential equations, together with boundary conditions for training an artificial neural network in order to solve the problem has been proposed by Lagaris et al. (1998). Recent developments related to automatic differentiation of the neural networks in order to approximate the required derivatives and its implementation with the usage of open-source software, led to the development of the physics-informed neural networks (see, among others, Raisi et al. 2019, Baydin et al. 2018, Shin et al. 2020, Kharazmi et al. 2019, Karniadakis et al. 2021). A PINN architecture requires the classical elements of feed-forward neural networks, like nodes, hidden layers, activation functions, and instead of the availability of input-output data for training, the physics law, with the governing differential equations and boundary conditions are used. Artificial neural networks with various hidden layers, which exponentially reduce the computational cost and amount of training data in some applications along others have been proposed by Al-Aradi et al. (2018).

The PINNs combine a collocation approach for fitting the solution of the governing differential equations and boundary conditions at certain points of a domain and it's boundary. After using optimization techniques the parameters of the neural networks, weights and biases, are found. The backpropagation improves the results on each epoch of training, till convergence.

In the present paper PINNs are applied to Kirchhoff plate bending problems with transverse distributed loads and to contact models with elastic Winkler's foundations. The Adam's optimization algorithm (Kingma and Ba 2015), based on gradient decent method and more commonly used in deep learning together with the L-BFGS (Limited- Broyden–Fletcher–Goldfarb–Shanno) technique (Fletcher 1987) are applied in this work. The L-BFGS optimization is intended for solving unconstrained nonlinear optimization problems. It is widely used for large scale problems and for parameter estimation in machine learning.

The paper is organized as follows. In Section 2 the plate bending and contact problems are described. The Kirchhoff governing equations with transverse forces and transverse-foundation reaction forces are formulated. The neural network architecture for the plate models is presented in Section 3. The scheme, based on calculating error loss function, i.e. errors for the fitting the governing equations and the boundary conditions at collocation points is presented. The computational algorithm involving the feedforward and backpropagation steps are described in Section 4. Numerical results are presented in Section 5. Conclusions and discussions are provided in Section 6.

2. The plate bending and the contact problems

The classical Kirchhoff elastic plate bending equation reads,

$$D\Delta^2 W(x,y) = F(x,y), x,y \in G, \tag{1}$$

where $G = (0,l_1) \times (0,l_2)$, l_1, l_2 are the lengths of sides of the plate, $W(x,y)$ denotes the deflection, $D = h^3E/12(1-\nu^2)$ is the cylindrical rigidity of the plate, h is the thickness of the plate, E is the Young modulus and ν is the Poisson ratio. Further, Δ^2 is the biharmonic operator and $F(x,y)$ are the transverse distributed loading forces.

The simplest soil structure interaction effect is described by the classical linear Winkler's foundation in which the soil reaction on the plate is directly proportional to the deformation of the plate. Another type of Winkler's foundation is a separated nonlinear elastic Winkler-type foundation spring with contribution $k_1W - k_2W^3$ (k_1 and k_2 are Winkler-type stiffness's constants). This subgrade model, often used in practice for nonlinear elastic foundations (e.g. Katsikadelis and Yiotis 2003, Shen Hui-Shen 1995), is investigated here. Namely, the following equation is considered,

$$LW + p(W) = F, \quad x,y \in G, \tag{2}$$

where $LW \equiv D\Delta^2 W$ and $p(W) = k_1W - k_2W^3$. The proposed neural network learning techniques can be applied to different kind of boundary conditions for plates (see, e.g. Muradova and Stavroulakis 2020). In the examples in Section 5, training of the neural network is performed for the simply supported and clamped plate.

3. Neural network architecture

The artificial neural network architecture, used in this work is illustrated in Figure 1.

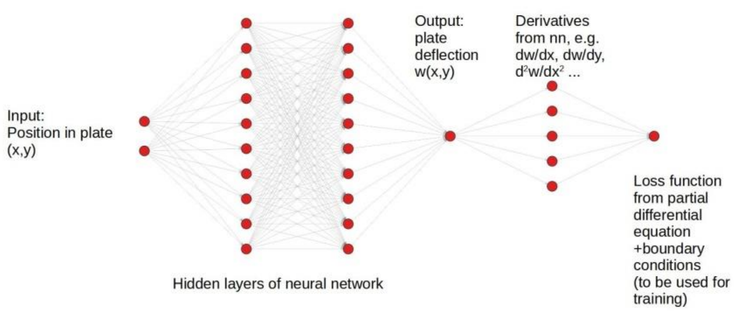

Fig. 1. The physics-informed artificial neural network architecture.

The input data are the position vector x, y and the output of the neural network is an approximate solution $\overline{W}$. The mesh consisting from the collocation points for the equations (1), (2) and for the boundary ∂G is $\overline{G}_{N_1N_2} = \{(x_i, y_j), i = 0,1, \dots N_1+1, j = 0,1,\dots,N_2+1, x_0 = 0, x_{N_1+1} = l_1, y_0 = 0, y_{N_2+1} = l_2\}$ on the plate is considered. In case, $N_1 = N_2 = N$ the number of collocation (interior) points for the equation (1) and (2) is $N \times N$. The parameters of the neural networks W, F for the equations (1) and (2) with boundary conditions are learned by minimizing the mean squared error, named loss function,

$$MSE = MSE_b + MSE_f.$$

Here MSE_b is the loss for boundary conditions

$$MSE_{B_k} = \frac{1}{N_b}\sum_{i,j=0}^{N_b-1}\left\{\left(B_kW(0,y_j) - W_{0j}^k\right)^2 + \left(B_kW(l_1,y_j) - W_{1j}^k\right)^2 + \left(B_kW(x_i,0) - W_{i0}^k\right)^2 + \left(B_kW(x_i,l_2) - W_{i1}^k\right)^2\right\},$$

where B_k the corresponding differential operator, and MSE_F is the loss for $LW + p(W) - F(x,y)$, i.e.

$$MSE_F = \frac{1}{N}\sum_{i,j=1}^{N}\left\{LW(x_i,y_j) + p\left(W(x_i,y_j)\right) - F(x_i,y_j)\right\}^2,$$

where $N_b = N + 2$ is the number of collocation points on the boundary and N is the number of collocation points for the equations (1) and (2). For example, the case $k = 0,2$, $B_0W = W, B_2W = \Delta W$ with

$$W_{0j}^k = 0,\ W_{1j}^k = 0,\ W_{i0}^k = 0,\ W_{i1}^k = 0,\quad x,y \in \partial G$$

corresponds to the simply supported plate,

$$W = \Delta W(x,y) = 0, x,y \in \partial G, \tag{3}$$

and when $k = 0,1$, $B_0W = W, B_1W = \partial W/\partial n$ we have the clamped boundary conditions,

$$W_x(0,y) = 0,\ W_x(l_1,y) = 0,\quad W_y(x,0) = 0,\ W_y(x,l_2) = 0,\ x,y \in \partial G. \tag{4}$$

4. Feedforward and backpropagation of the neural network

The computational algorithm is based on feedforward of the signal through the neural network and backpropagation of the error based on the loss function. During backpropagation the weights are updated and eventually training is performed.

Step 1. Give the input data for the neural network, the vectors x, y (the collocation points for the equations, i.e. $x = (x_1, x_2 \dots, x_N), y = (y_1, y_2 \dots, y_N)$ and the collocation points for the boundary conditions $x = (x_0, x_1, \dots, x_N, x_{N+1})$, $y = (y_0, y_1, \dots, y_N, y_{N+1})$, where x_0, y_0, x_{N+1} and y_{N+1} are also included.

Step 2. Set up the physical parameters of the plate, the loading function and in case of the presence of the elastic foundations the coefficients k_1, k_2. In feedforward, for each neuron of the first hidden layer we compute $z_k^{ij} = x_iw_k^1 + y_jw_k^2 + b_k$, $i,j = 0,1,\dots,N+1$ (for the boundary points) and $i,j = 1,2,\dots,N$ (for the interior points), $k = 1,2,\dots,H$, H is the number of neurons in the hidden layers. Here w_k^1, w_k^2 are the weights from the input layer x, y, respectively to the neurons of the first hidden layer and b_k are the biases.

Step 3. Compute an activation function for the neural network (Sigmoid, Tanh, ReLU etc). In case of using the sigmoid activation function for the first hidden layer for each neuron k, $S_{1k} = \sigma(z_k^{ij})$, $\sigma(z_k^{ij}) = 1/\left(1 + e^{-z_k^{ij}}\right)$. Then for the second hidden layer compute

$$S_{2k} = \sigma\left(\sum_{l=1}^{H}\sigma\left(z_l^{ij}\right)w_{lk} + b_k\right) = \sigma\left(\sum_{l=1}^{H}S_{1l}w_{lk} + b_k\right)$$

from the previous hidden layer and so on for all the rest hidden layers compute $S_{pk}, p = 3,4,\dots N_h$ (N_h is the number of hidden layers).

Step 4. Calculate the output of the neural network,

$$\overline{W}_{ij} = \sum_{k=1}^{H} S_{N_h k} v_k + b,$$

where $\overline{W}_{ij} = \overline{W}(x_i, y_j)$, v_k are the weights from the last hidden layer to the output layer and b is the bias. The output is the approximate solution of the equation (1) or (2) with the chosen boundary conditions.

The gradients are computed along with the approximate solution and then the loss function is calculated with the use of tools of Tensorflow scientific package. In backpropagation the weights and biases are updated and the loss function is minimized by applying Adam's and the L-BFGS-B optimizers of Tensorflow, respectively. The output of the neural network is compared with the exact solution and with the numerical solution, obtained after using the spectral collocation method.

5. Numerical results

Below in the numerical examples the neural network is trained for the simply supported and clamped plates. The programming code for implementation of the techniques is composed in Python version 3.7.9, the platform for machine learning, Tensorflow 1.14.0 in Python language. In addition, NumPy (the core library for scientific computing) and Matplotlib (the plotting library) are used in the programming code. The sigmoid activation function is used, since it is a differentiable function and suitable for automatic differentiation. The Adam's and L-BFGS-B optimizers have been tested for updating the weights and minimization of loss functions. In Adam's optimizer: learning_rate=0.001, and $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-08$. In L-BFGS-B's optimizer maxiter=50000, 'maxfun': 50000, 'maxcor': 50, 'maxls': 50, 'ftol': 1.0 * np.finfo(float).eps).

For estimation of the error of approximation the following notation is introduced: $\|W\| = max_{ij}\{|W_{ij}|\}$.

Example 1. (The simply supported plate)

The physical parameters of the plate are $D = 1, l_1 = l_2 = 1$, and the transverse forces are $F(x,y) = \sin(\pi x)\sin(\pi y)$. The exact solution is $W(x,y) = (1/16\pi^4)\sin(\pi x)\sin(\pi y)$. The neural network consists from 4 hidden layers with 100 neurons, two inputs **x** and **y** and output $\overline{W}$. For training the neural network, $N = 16$, epochs 50000, 80000 have been tested. The numerical results for the approximate solution $\overline{W}$ for (1), (3) along a cross-section of the plate are shown on Figure 2.

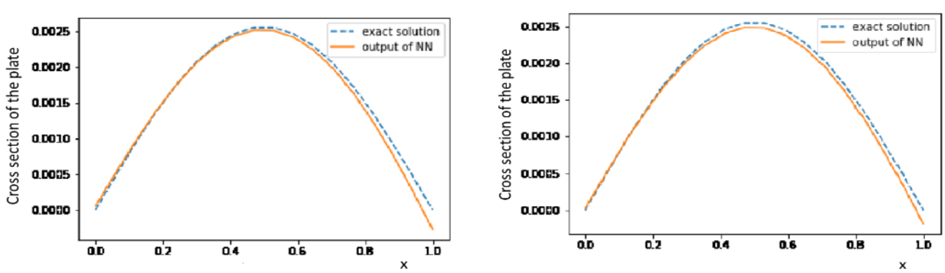

Fig. 2. The cross section of the approximate solution $\overline{W}$ with 256 interior collocation points (a) epochs are 50000; b) epochs are 80000).

The norm of the error of the approximate vector solution is $\varepsilon = \|W - \overline{W}\| = 0.0006309$ (W is the exact solution) with epochs 50000. For epochs 80000 we have obtained $\varepsilon = 0.0004778$. Table 1 shows dependence of the error of the output of the PINN on the number of hidden layers and neurons. In Table 1 it has been taken N=10 and epochs=150000.

Layers \ Neurons	**10**	**20**	**40**
2	$1.13313 \cdot 10^{-3}$	$7.04430 \cdot 10^{-4}$	$4.56145 \cdot 10^{-4}$
4	$5.65454 \cdot 10^{-4}$	$5.70129 \cdot 10^{-4}$	$4.56497 \cdot 10^{-4}$
6	$6.03632 \cdot 10^{-4}$	$5.55525 \cdot 10^{-4}$	$1.87982 \cdot 10^{-4}$

Table 1. The norm of the error of the approximate vector solution ε.

The time of computations increases with increasing the number of layers, the number of neurons and the number of collocation points. Figure 3 shows dependence of the error loss function on the number of collocation points. A dynamics of changing of the error loss function with respect to the epochs is illustrated.

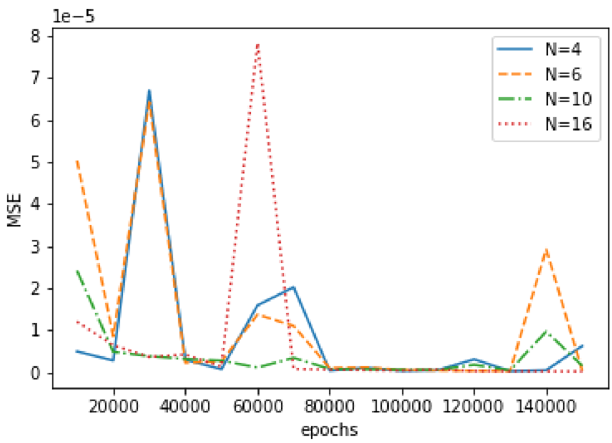

Fig. 3. The error loss function MSE in correspondence with epochs.

For the same physical parameters and transverse forces of the plate we have trained the neural network for the problem (2) with the foundations' parameters $k_1 = 30, k_2 = 1$. The number of epochs are 500000, the hidden layers are 4 with 40 neurons. The interior collocation points are 256 and the boundary collocation points $N_b = 18$. The results are shown on Figure 4 (W is the exact solution with $k_1 = 0, k_2 = 0$). The results are compared with the spectral collocation method (Muradova et al. 2018). The analytical solution is expanded into the partial sums of the double Fourier series, i.e. the displacement at the collocation points is computed as

$$W_{ij}^{sp} = \sum_{k,l=1}^{M} \phi_{kl} \sin\left(\pi k \frac{x_i}{l_1}\right) \sin\left(\pi \frac{y_j}{l_2}\right),$$

where $M = 16$ ($M \times M$ is the number of global basis functions) and $i, j = 1,2, \dots, N$. The global basis functions satisfy the boundary conditions (3).

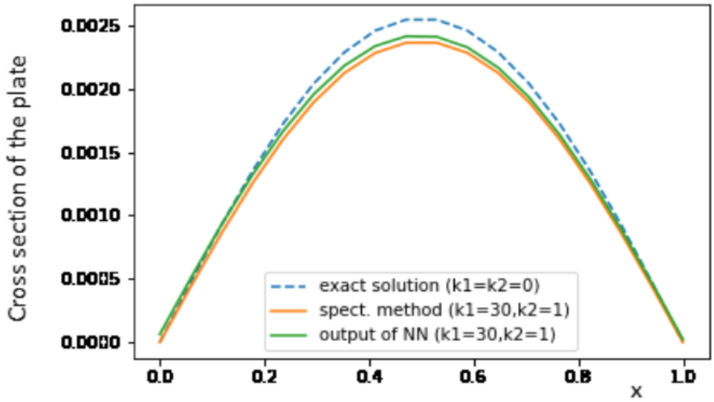

Fig. 4. The cross section of the approximate solutions with $k_1 = 30, k_2 = 1$ (after applying the spectral collocation method and training of the PINN, respectively) and the exact solution with $k_1 = 0, k_2 = 0$.

Example 2. (The clamped boundary conditions.)

The physical parameters of the plate $D = 1, l_1 = l_2 = 1$, and the transverse forces of the plate $F(x, y) =$cos2πx(cos2πy-1)+(cos2πx-1)cos2πx are considered. The exact solution with $k_1 = 0, k_2 = 0$ is $W(x, y) = 1/16\,\pi^4$(cos2πx-1)(cos2πy-1). We have trained the neural network for the unconstrained problem (1) and the constrained problem (2) with the clamped boundary conditions (4). The hidden layers for both problems are 2 with 40 neurons. The collocation points including interior and boundary are 484($N = 22$). For (1), i.e. when $k_1 = 0, k_2 = 0$ the number of epochs is 500000. The results are shown on Figure 5.

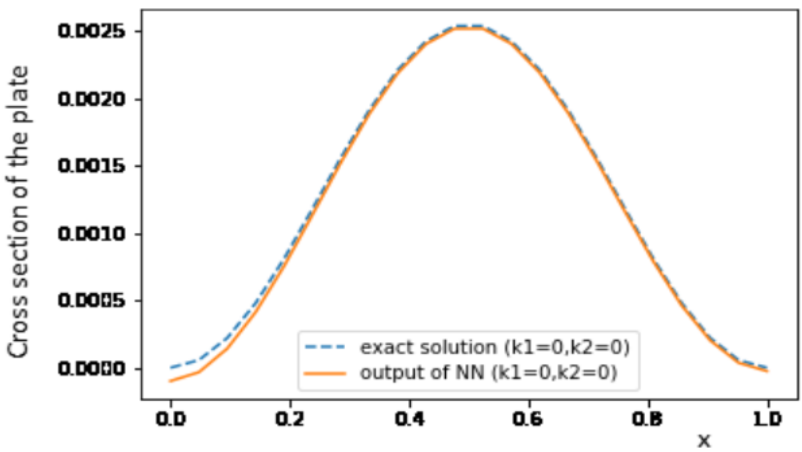

Fig. 5. The cross section of the exact solution and the approximate solutions with $k_1 = 0, k_2 = 0$ after training the PINN, respectively.

The constrained problem (2) has been solved for different values of the foundation constants k_1 and k_2 as well. Figure 6. shows how the loss error function MSE for different values of the foundation constants k_1 and k_2 changes with increasing the number of epochs.

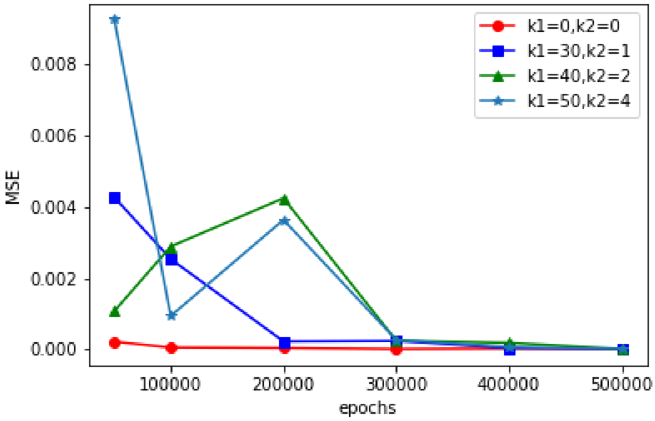

Fig. 6. The error loss function MSE for different values of the foundation constants k_1 and k_2 in correspondence with epochs.

From the numerical results we can see that quite good approximations can be reached with just 2 hidden layers. Numerous numerical experiments have shown that with increasing mostly the first Winkler-type stiffness's constant in order to reach good accuracy of the approximation quite many training iterations must be performed. Other words, in order to the NN could learn the behavior of the solution it is needed a large number of training epochs.

6. Conclusions

The Kirchhoff plate model with transverse loading forces and with foundation interaction forces have been trained by artificial PINNs. Training of the neural network has been performed by feedforward and backpropagation using numerical optimizations as Adam's optimizer and the L-BFGS algorithm. The architecture of the PINN is composed, based on the nature of the problem. The steps of the computational algorithm are described.

The numerical results have been obtained for the simply supported and clamped boundary conditions. The efficiency of the proposed techniques has been shown on examples. The results of the output of the PINNs are compared with the approximate solution, obtained after the application of the spectral collocation method, and with the exact solution. The accuracy of the approximate solution mainly depends on the number of training iterations (epochs), the number of hidden layers with neurons and the number of collocation points (samples).

As the results have shown in order to reach high accuracy of approximation deep learning with many training iterations are required. However it should be taken into account that time of computations increases much with increasing a number of hidden layers. How many layers, neurons, collocation points and epochs must be taken depends on the type of the considered problem. Usually a complexity of the mechanical models requires more training steps for learning in the PINN.

The programming code has been composed, based on scientific python packages. The Tensorflow library of Python, intended for artificial neural networks, has been used. The programming code can easily be modified and used for other material-made plates and for plates with other types of boundary conditions such as, for example, partially clamped or boundary conditions for free edges of the plates. These are advantages of PINNs over traditional, for example, finite elements, boundary elements or spectral methods where local or global basis functions are needed to be defined. However, as the results have shown in order to reach a good approximation of a solution of mechanical problems many training iterations must be provided for neural networks.

PINNs, as already mentioned, are considered to be useful for the quick calculation of estimates in comparison to complicated classical methods. This advantage can be useful for certain real-time applications, for instance in robotics, virtual reality etc. Furthermore PINNs are being developed for the unified study of direct and inverse problems, which are known to be difficult to be solved by classical methods Karniadakis et al. (2021).

References

Al-Aradi A., Correia A., Naiff, D. Jardim G. and Saporito Y. (2018). Solving nonlinear and high-dimensional partial differential equations via deep Learning, arXiv:1811.08782.

Avdelas A.V., Panagiotopoulos P.D. and Kortesis S. (1995). Neural networks for computing in the elastoplastic analysis of structures, Meccanica, 30, 1–15.

Baydin A. G., Pearlmutter B. A., Radul A. A. and Siskind J. M. (2018). Automatic Differentiation in Machine Learning: a Survey, Journal of Machine Learning Research 18, 1-43. https://www.jmlr.org/papers/volume18/17-468/17-468.pdf

Fletcher R., Practical Methods of Optimization (2nd ed.) (1987). John Wiley & Sons, New York.

Kharazmi E., Zhang, Z. and Karniadakis G. (2019). Variational physics-informed neural networks for solving partial differential equations. Preprint at
arXiv https://arxiv.org/abs/1912.00873.

Karniadakis G.E., Kevrekidis, I.G., Lu, L. et al. (2021). Physics-informed machine learning. Nat Rev Phys 3, 422–440. https://doi.org/10.1038/s42254-021-00314-5

Katsikadelis J.T. and Yiotis A.J. (2003). The BEM for plates of variable thickness on nonlinear biparametric elastic foundation. An analog equation solution, Journal of Engineering Mathematics, 46, 313–330.

Kingma D. P. and Ba J. (2015). Adam: A Method for Stochastic Optimization, Computer Science, Mathematics, The International Conference on Learning Representations (ICLR).

Kortesis S. and Panagiotopoulos P.D. (1993). Neural networks for computing in structural analysis: Methods and prospects of applications, International Journal for Numerical Methods in Engineering, 36, 2305-2318.

Lagaris E., Likas A. and Fotiadis D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks, 9, 987–1000.

McCulloch W. S. and Pitts W. (1943). A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics, Vol. 5(4), 115-133.

Muradova A. D. (2008). The spectral method and numerical continuation algorithm for the von Kármán problem with postbuckling behaviour of solution, Advances in Computational Mathematics, Vol. 29(2), 179-206.

Muradova A. D. and Stavroulakis G. E. (2007). The projective-iterative method and neural network estimation for buckling of elastic plates in nonlinear theory, Communications in Nonlinear Science and Numerical Simulation, Vol. 12, 1068-1088.

Muradova A. D. and Stavroulakis G.E. (2012). Buckling and postbuckling analysis of rectangular plates resting on elastic foundations with the use of the spectral method, Computer Methods in Applied Mechanics and Engineering. Vol. 205–208, 213–220.

Muradova A. D., Stavroulakis G. E. and Tairidis G. K. (2018). A spectral collocation method for vibration suppression of smart elastic plates. CD-Book of Papers, 9th GRACM International Congress on Computational Mechanics, Chania, 352-360.

Muradova A. D. and Stavroulakis G. E. (2020) Mathematical Models with Buckling and Contact Phenomena for Elastic Plates: A Review. *Mathematics*, MDPI, *8(4)*, 566.

Raissi M., Perdikaris P. and Karniadakis G. E. (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.

Shen Hui-Shen (1995). Postbuckling analysis of orthotropic rectangular plates on nonlinear elastic foundations, Engineering Structures, 17 (6), 407–412.

Shin, Y., Darbon, J. and Karniadakis, G. E. (2020). On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. Commun. Comput. Phys. 28, 2042–2074.

Stavroulakis G.E. (2000). Inverse and Crack Identification Problems in Engineering Mechanics, Springer.

Stavroulakis G. E., Avdelas A. V., Abdalla K. M. and Panagiotopoulos P. D. (1997). A neural network approach to the modelling, calculation and identification of semi-rigid connections in steel structures, Journal of Constructional Steel Research, Vol. 44(1–2), 91-105.

Stavroulakis G., Bolzon G, Waszczyszyn Z. and Ziemianski L. (2003). Inverse analysis. In: Karihaloo B, Ritchie RO, Milne I (eds in chief) Comprehensive structural integrity, Numerical and computational methods, Vol 3, Chap 13. Elsevier, Amsterdam, 685–718.

Waszczyszyn Z. and Ziemiański L. (2005). Neural Networks in the Identification Analysis of Structural Mechanics Problems. In: Mróz Z., Stavroulakis G.E. (eds) Parameter Identification of Materials and Structures, CISM International Centre for Mechanical Sciences (Courses and Lectures), vol 469. Springer, Vienna.

Yagawa G, and Oishi A. (2021). Computational mechanics with neural networks. Springer.