Solving Plane Elasticity with an Ensemble of Physics-informed Neural Networks

Aliki D. Mouratidou[1], Georgios A. Drosopoulos[2], Georgios T. Stavroulakis[1,*]

[1]Technical University of Crete, Chania,Creece; [2]University of Central Lancashire, Preston, UK;
[*]Corresponding author: gestavroulakis@tuc.gr.

Keywords: Elasticity, partial differential equations, artificial intelligence, physics-informed neural networks.

1. Introduction

Elastic structures in solid mechanics are simulated via an ensemble of physics-informed neural networks. The proposed computational approach is based on principles of artificial intelligence. A deep learning is performed through training the PINN model in order to fit the elasticity equations and associated boundary conditions at collocation points, without need of input-output data. An open source machine learning platform is used, based on Tensorflow, written in Python and Keras library, an application programming interface, intended for a deep learning.

Artificial neural networks (ANNs) have led to revolutionary advances in the manufacturing industry. The main features of a neural network is an architecture which identifies the connections between layers and neurons, dataset which consists of training, validation and testing data, an optimization algorithm for minimizing the loss function and updating the weights and biases between the neurons.

Physics-informed neural networks (PINNs) is relatively recent development. It is most promising research direction in computational artificial intelligence, which can simulate and solve scientific and engineering problems involving differential equations thanks to automatic differentiation of the neural network metamodel and modern open-source scientific software. PINNs have been applied in a wide range of scientific computing applications (Karniadakis et al.[1], Raissi et al. [2], Baydin et al. [3], Lagaris et al. [4], Meade and Fernadez [5], Shin et al.[6], Haghighat et al.[7], Muradova and Stavroulakis [8], Katsikis et al.[9], Cai et al.[10], Chen et al.[11], Guo and Haghighat [12]).

2. Elastic Model Description

The equations of elasticity consist of equilibrium equations of stresses and loads, constitutive equations that relate stresses and strains (Hooke's law), and strain-displacement relations, i.e. for 2-D linear elasticity case:

$$\sigma_{xx,x} + \sigma_{xy,y} + f_x = 0, \sigma_{xy,x} + \sigma_{yy,y} + f_y = 0, \quad (1)$$

$$\sigma_{xx} = \lambda(\varepsilon_{xx} + \varepsilon_{yy}) + 2\mu\varepsilon_{xx},\ \sigma_{yy} = \lambda(\varepsilon_{xx} + \varepsilon_{yy}) + 2\mu\varepsilon_{yy},\quad \sigma_{xy} = 2\mu\varepsilon_{xy,} \quad (2)$$

$$\varepsilon_{xx} = u_{x,x},\ \varepsilon_{yy} = u_{y,y},\ \varepsilon_{xy} = \frac{1}{2}(u_{x,y} + u_{y,x}), \quad (3)$$

where σ_{xx}, σ_{yy} and σ_{xy} are the components of the stress tensor,$\sigma_{xx,x}$,$\sigma_{yy,y}$,$\sigma_{xy,y}$, $\sigma_{xy,x}$, are their derivatives with respect to *x* or *y*, respectively f_x, f_y are the external body forces, ε_{xx}, ε_{yy} and ε_{xy} are the components of the strain tensor, and λ and μ are the Lamé parameters. Further, u_x and u_y are the displacements in *x* and *y* directions, respectively, and their derivatives on *x* or *y* are $u_{x,x}$, $u_{x,y}$, $u_{y,y}$, $u_{y,x}$. In the above formulas, *x* and *y* are the Cartesian coordinates and $(x, y) \in \Omega$, where Ω is the domain of definition for the stresses, strains and displacements.

3. Architecture of the Physics-informed Multi-neural Network Model

A neural network consists of input, hidden and output layers. Each layer provides neurons which are connected with the neurons from the previous and next layers. A training of a network is the most important part of machine learning since through this process a loss function (prediction error of the neural network) is minimized and the weights and biases are updated. The training is performed through feedforward and backpropagation processes.The weights and biases are optimized by using the Adam's optimization algorithm.

For the elasticity system (1)-(3) the proposed PINN model consists of two types of neural networks: surrogate networks and a residual network. The surrogate neural networks are intended for computations of the components of the stress and strain tensors, and the displacement field, respectively. The surrogate networks take as input the collocation points where the elasticity equations and associated boundary conditions are fitted. The architecture of the multi-PINN, developed in this work, is presented in **Figure 1**.

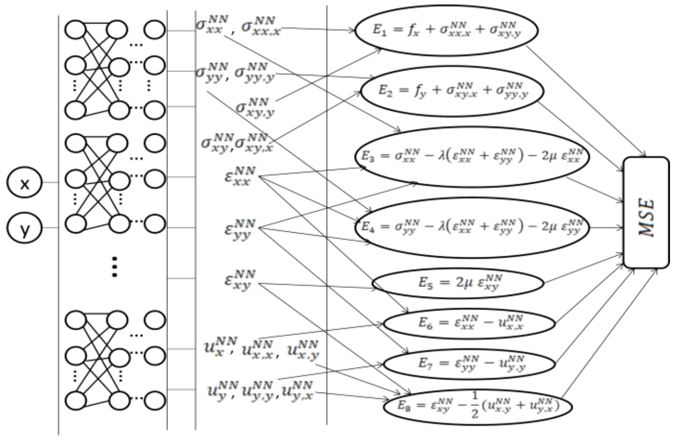

Figure 1. Architecture of the multi-PINN with eight surrogate NNs.

In **Figure 1** there are two inputs $(\mathbf{x}, \mathbf{y})$, $\mathbf{x} = (x_0, x_1, \ldots, x_N), \mathbf{y} = (y_0, y_1, \ldots, y_N)$ and MSE is the mean square error, i.e. the loss function for the problem, defined as $MSE = MSE_e + MSE_b$,whereMSE_e is the loss function for (1)-(3), and MSE_b is the loss error for the associated boundary conditions. If one denotes by $E_{k,ij} = E_k(x_i, y_j)$, then

$$MSE_e = \frac{1}{N2}\sum_{i,j=1}^{N}\left(\sum_{k=1}^{8} E_{k,ij}^2\right),\ MSE_b = \frac{1}{N}\sum_{i=1}^{N}\left(E_{k,0i}^2 + E_{k,Ni}^2 + +E_{k,i0}^2 + E_{k,iN}^2\right).$$

In the surrogate models 1,2,3,7,8, for the stress tensor and displacements automatic differentiation is used. The MSE is minimized within the backpropagation algorithm by applying the Adam method.

4. Numerical results

In this section we solve (1)-(3) by the multi-PINN for a rectangular plane elasticity structure with boundary conditions for the displacements and the components of the stress tensor.

The rectangular shape is, $0 \le x \le l_1, 0 \le y \le l_2$, where l_1,,l_2 are lengths of the sides of the structure. The structure is fixed on the *x*=0 side, $0 \le y \le l_2$, under vertical loading external constant forces on the other parallel side *x*=l_1 and the other parallel sides (*y*=0 and *y*=l_2) are free (see **Figure 2**). Then on the loading edge the component of the stress tensor in *x* direction $\sigma_{xx} = 0$. On the free edges $\sigma_{xx} = 0$ and $\sigma_{yy} = 0$. Thus, the boundary conditions are $u_x(0, y) = 0,\ u_y(0, y) = 0,\ \sigma_{xy}(l_1, y) = P,\ (P \neq 0), \sigma_{xx}(l_1, y) = 0, \sigma_{xx}(x, 0) = 0,\ \sigma_{yy}(x, 0) = 0,\ \sigma_{xx}(x, l_2) = 0,\ \sigma_{yy}(x, l_2) = 0$.

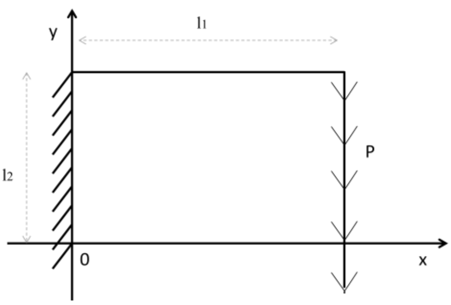

Figure 2. Plane elasticity structure with fixed and loading edges.

The numerical experiments have been done with lengths $l_1 = 1.82, l_2 = 1.69$. The module of elasticity E=200000, the Poisson ratio $\nu = 0.3$ and the Lame parameters are$\lambda = E\nu/((1+\nu)(1-2\nu))$, $\mu = E/(2(1+\nu))$. The number of training samples is 21x21, test samples is 15x15. The number of epochs is 2000 with the error $< 10^{-4}$. In **Figure 3a)** the loss error with respect to epochs is plotted and in **Figure 3b**) the structure, obtained after the training of the multi-PINN is shown.

a) b)

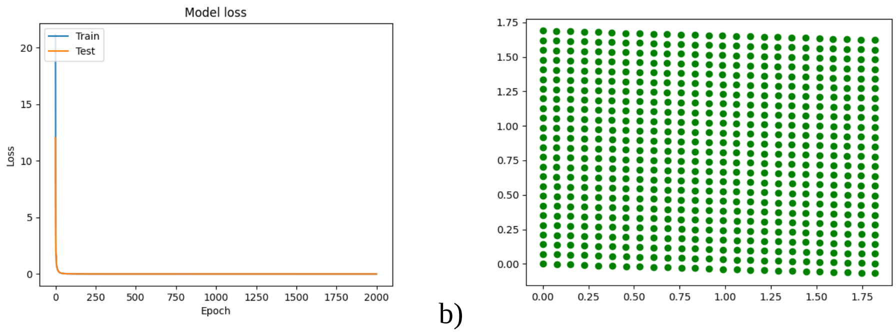

Figure 3.a) Model loss with respect to epochs; b)Structure, obtained after the training the PINN.

The numerical experiments have shown that the accuracy of the computations depends mostly on the number of samples (collocation points), the number of epochs and the number of layers and neurons.

The proposed technique can be extended and applied to linear and nonlinear, direct and inverse problems in engineering and manufacturing.

References

1. Karniadakis G.E., Kevrekidis I.G., Lu L. et al.: Physics-informed machine learning. Nat Rev Phys, 3, 422–440 (2021).
2. Raissi M., Perdikaris P., Karniadakis G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707 (2019).
3. Baydin A., Pearlmutter B. A., Radul A. A., Siskind J. M. Automatic Differentiation in Machine Learning: a Survey(2018), https://arxiv.org/pdf/1502.05767.pdf
4. Lagaris E., Likas A., Fotiadis D. I. Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks, 9, 987–1000 (1998).
5. Meade A. J., Fernandez A. A.: The numerical solution of linear ordinary differentialequa-tions by feed-forward neural networks, Mathematical and Computer Modeling, 19, 1–25 (1994).
6. Shin, Y., Darbon, J., Karniadakis, G. E.: On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. Commun. Comput. Phys. 28, 2042–2074 (2020).
7. Haghighat, E., Raissi M., Moure A., Gomez H., and Juanes, R. A deep learning frame-work for solution and discovery in solid mechanics, https://arxiv.org/abs/2003.02751 (2020)
8. Muradova A. D., Stavroulakis G. E. Physics-informed neural networks for elastic plate problems with bending and Winkler-type contact effects. Journal of the Serbian Society for Computational Mechanics, 15(2), 45-54 (2021).
9. Katsikis D., Muradova A. D. and Stavroulakis G. S.: A Gentle Introduction to Physics-Informed Neural Networks, with Applications in Static Rod and Beam Problems, Journal of Advances in Applied \& Computational Mathematics 9, 103-128 (2022).
10. Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E. Physics-informed neural networks (pinns) for fluid mechanics: A review. ActaMechanicaSinica (2022), 1–12.
11. Chen, Y., Lu, L., Karniadakis, G. E., and Dal Negro, L.: Physics-informed neural networks for inverse problems in nano-optics and metamaterials. Optics express 28(8),11618–11633 (2020).
12. Guo M., Haghighat E. An energy-based error bound of physics-informed neural network solutions in elasticity. arXiv preprint arXiv:2010.09088(2020).