A Neural Network-based Approach for Bending Analysis of Strain Gradient Nanoplates

C.A. Yan[1], R. Vescovini[1]*and N. Fantuzzi[2]

[1]*Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano*

Via La Masa 34, 20156 Milano, Italy

[2] *Dipartimento di Ingegneria Civile, Chimica, Ambientale e dei Materiali - Università di Bologna*

Viale del Risorgimento 2, 40136 Bologna, Italy

Abstract

Purpose of this paper is the presentation of a novel Machine Learning (ML) technique for nanoscopic study of thin nanoplates. The second-order strain gradient theory is used to derive the governing equations and account for size effects. The ML framework is based on Physics-Informed Neural Networks (PINNs), a new concept of Artificial Neural Networks (ANNs) enriched with the mathematical model of the problem. Training of PINNs is performed using a highly efficient learning algorithm, known as Extreme Learning Machine (ELM). Two applications of this ANNs-based method are illustrated: solution of the Partial Differential Equations (PDEs) modeling the flexural response of thin nanoplates (direct problem), and identification of the length scale parameter of the nanoplate mathematical model with the aid of measurement data (inverse problem). Comparison with analytical and Finite Element (FE) solutions demonstrate the accuracy and efficiency of this ML framework as meshfree solver of high-order PDEs. The stability and reliability of the present method are verified through parameter studies on hyperparameters, network architectures, data noise and training initializations. The results presented give evidence of the effectiveness and robustness of this new ML approach for solving both direct and inverse nanoplate problems.

Keywords: Physics-Informed Neural Networks, Extreme Learning Machine, Nanoplates, Bending Analysis, Strain Gradient Theory, Parameter Identification.

---

*Corresponding author. *Email address:* `riccardo.vescovini@polimi.it` (Riccardo Vescovini)

1 Introduction

Over the past years, a growing interest has been devoted to the research field of nanotechnology. Nanomaterials and nanostructures are employed to fabricate microelectromechanical (MEMS) and nanoelectromechanical (NEMS) systems which find a large range of applications in sectors like mechanical, aerospace, electronic and biomedical engineering. Some examples of micro- and nano-sized systems include biosensors, electrochemical actuators, supercapacitors, nanotubes and nanoprobes [1–4], which largely make use of nanorods, nanobeams, nanoplates and nanoshells as fundamental structural units [5].

In order to effectively analyze and design MEMS and NEMS, a large number of experimental and theoretical studies have been conducted to characterize the mechanical behavior of nanostructures. Recent experimental works have demonstrated the importance of material microstructures and long-range interaction forces on the mechanical response of micro- and nano-structural members [6–8]. These small-scale effects cannot be captured properly by Classical Continuum Theories (CCTs) which are widely used for modeling structures at macroscopic level.

Therefore, the recent scientific literature has seen a drastic surge in studies focused on the development of new theories for modeling nanostructures. Among the different modeling possibilities, a viable approach is represented by a direct modification of CCTs to give the so-called Nonlocal Continuum Theories (NCTs). Several NCTs have been proposed by researchers throughout the years. Some classical examples are given by the stress gradient, strain gradient, couple stress and integral types theories [9–12], where the small scale effects are taken into account through the introduction of one or more internal length scale parameters. Recently, other size-dependent elasticity theories have emerged to improve the existing ones, such as the modified couple stress theory due to Yang et al. [13], which involves only one length scale parameter for representing the microstructural effects. The nonlocal strain gradient theory developed by Aifantis [14] puts together the stress gradient and strain gradient theories. Two size-dependent lengths are adopted in this case to include strain and stress gradients on the constitutive response of materials. The modified strain gradient theory proposed by Lam [7] uses three internal length parameters to reflect the dilatation gradients, deviatoric stretch gradients, and rotation gradients. The recently developed nonlocal modified gradient theory presented by Faghidian [15] adopts instead four intrinsic length scales to capture the effects of nonlocality, dilatation gradient, deviatoric stretch gradient, and symmetric rotation gradient. Higher-order nonlocal gradient theories have also been proposed lately. Some examples are given by the two-phase local/nonlocal gradient model [16], which adopts five intrinsic parameters (two nonlocal, one mixture, two gradient length scale parameters), and the higher-order unified gradient elasticity theory in [17], where two strain gradient characteristic lengths and two stress length scale parameters are introduced to model nano-scale effects. Among these theories, the second-order strain gradient theory has gained popularity over the years due to its formulation simplicity. Indeed, only one additional material parameter is introduced into the constitutive law to account for size effects [18].

Differently from CCTs, the governing equations derived using NCTs are characterized by a higher differential order. This is due to the presence of higher-order deformation and force terms introduced by the length scale parameters. In the context of thin nanoplates modeled with the second-order strain gradient theory, the partial differential equations describing the flexural equilibrium pass from a differential order of four to an order of six. This brings a significant challenge to traditional solution methods.

Ad-hoc analytical approaches based on Navier and Levy methods were proposed in the literature. Papargyri-Beskou and Beskou [19] derived closed-form solutions for isotropic thin nanoplates under simply supported boundary conditions. The Navier method was adopted in the context of the second-order negative strain gradient theory to find analytical solutions for displacements, buckling loads and natural frequencies. Babu and Patel [18] used a procedure based on Levy's approach to analyze isotropic nanoplates under static bending loading. They considered both classical and non-classical boundary conditions and compared their solutions with finite elements simulations. Cornacchia et al. [20] derived exact solutions for laminated composite nanoplates with different layouts. In this work, new analytical solutions were derived for both isotropic and antisymmetric orthotropic laminated nanoplates under static loading conditions. Akgöz and Civalek [21] presented analytical solutions based on Fouries series for bending, buckling and vibration analysis of micro-scale plates. The modified strain gradient elasticity theory was used in the study to capture the size-dependant effects. A semi-analytical approach was employed by Dastjerdi and Akgöz [22] to solve nanoscale problems. Their method was applied in the context of nanoplates modelled with the nonlocal theory of Eringen.

To overcome the inherent restrictions of analytical methods on geometry, material and boundary/loading conditions, numerical approaches such as Finite Element (FE) methods have also been proposed. To the best of the authors' knowledge, Babu and Patel [23] were the first to present a FE formulation incorporating second-order strain gradient effects. They compared two different formulations based on conforming and nonconforming elements for the static, stability and vibration analysis of isotropic nanoplates. An extension of this work to laminated nanoplates was presented by Bacciochi et al. [24]. In their work, the authors provided a comprehensive study on the numerical stability and reliability of the proposed FE formulation in predicting displacements and stress quantities under the effect of the length scale parameter. Other FE-based numerical approaches to solve the strain gradient equations of thin nanoplates can be found in the works of Niiranen et al. [25], who presented a method based on Isogeometric Analysis (IGA), and of Zhang et al. [26], who developed a framework combining the Differential Quadrature (DQ) and FE methods.

Solution methods based on meshless approaches [27] were also proposed in the recent literature as a potential alternative to FE-based approaches. Fabbrocino et al. [28] used the Radial Point Interpolation Method (RPIM) for solving the static bending analysis of Kirchhoff nanoplates. They conducted different parametric studies to test the influence of shape parameters on the accuracy of their numerical results. In a series of works, Wang et al. [29, 30] applied the Meshfree Galerkin method to analyze strain gradient thin plates. Both plates with rectangular and irregular shapes were considered in the simulations. Roque and Żur [31]

proposed the Radial Basis Function Finite Difference (RBF-FD) method to study the bending, free vibration, and transient responses of micro-plates with different boundary conditions and materials. Another possible mesh-free method for solving nanoscale problems is the Meshless Local Petrov–Galerkin (MLPG) method, which is used by Khorasani and Bayat [32] in the context of the modified couple stress theory.

Aside of this, an emerging trend regards the application of Physics-Informed Neural Networks (PINNs) as an effective mean for solving differential problems [33]. PINNs are an inherently meshless strategy relying upon the powerful framework offered by modern Artificial Neural Networks (ANNs). This new concept of ANNs have already been applied in different fields of science, such as fluid dynamics, quantum mechanics and solid mechanics [33–35], to cite a few. Recently the authors have proposed their applications for studying composite plate and shell structures at macroscale level [36]. Other recent contributions regarding the use of PINNs to plate problems can be found in [37–39].

In this work, the PINN framework presented in [36] is extended and applied, for the first time, to the static analysis of nanoplates. Specifically, the differential equations derived from the second-order negative strain gradient NCT are solved by means of PINNs. Differently from other existing PINN algorithms which are mainly based on Gradient-Based Learning (GBL) methods [37–39], here training of the networks is conducted under the paradigm of the Extreme Learning Machine (ELM) [40]. This new training approach provides much faster learning speed compared with classical learning strategies [41]. The superior efficiency of the PINN-ELM framework will be exploited to perform parametric studies. These will be useful to gain insights into the features of the method, including 1) the effect of different hyperparameters and network architectures on the direct solution of the governing equations, and 2) the possibility of carrying out model parameters identification in the context of inverse problems. The study illustrates the potential of this new meshless approach to handle a large variety of nanoplate problems, including analysis and model identification.

The paper is organized as follows: Section 2 gives an overview of the theoretical framework. The governing equations of strain gradient thin plates are derived. Direct and inverse solution of these equations are then discussed in the context of PINNs and ELM; Numerical results are shown in Section 3. Finally, Section 4 gives a summary of this work with suggestion for future developments.

2   Formulation

The definition of the governing equations represents the first step to define a PINN. In the following, the Partial Differential Equations (PDEs) used for modeling thin rectangular nanoplates are presented. The formulation is based on the second-order negative strain gradient nonlocal theory [10] under the Kirchhoff flexural plate hypothesis and linear elasticity assumptions [42]. This mathematical model is then exploited in the training process of PINNs for the solution and parameters identification of the relevant PDEs.

2.1 Governing equations: second-order negative strain gradient nonlocal theory

The convention adopted in the present work is schematized in Figure 1. The nanoplate is described using a Cartesian reference system with origin O taken on the middle surface. The three axes x, y and z are directed along the plate width a, length b and thickness h directions, respectively, while the corresponding displacement components are denoted as u_x, u_y and u_z.

The static equilibrium equations are described by the following PDE [42]:

$$\frac{\partial^2 M_{xx}}{\partial x^2} + 2\frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_{yy}}{\partial y^2} + q = 0 \tag{1}$$

where the forcing term is represented by the constant pressure load q, while M_{ij} are the moment resultants defined as:

$$M_{ij} = \int_{-h/2}^{h/2} \sigma_{ij} z \, dz \quad \text{for} \quad i, j = x, y \tag{2}$$

with σ_{ij} being the components of the stress tensor.

In the context of Kirchhoff's thin plate theory, the displacement field is given by:

$$\begin{cases} u_x(x, y, z) = -z\frac{\partial w(x,y)}{\partial x} \\ u_y(x, y, z) = -z\frac{\partial w(x,y)}{\partial y} \\ u_z(x, y, z) = w(x, y) \end{cases} \tag{3}$$

where w is the out-of-plane deflection of a point on the midsurface $z = 0$. Consequently, the linear strain-displacement relations read:

$$\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = z \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ -2\frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \tag{4}$$

with ϵ_{xx}, ϵ_{yy} and γ_{xy} defined as the in-plane components of the strain tensor.

According to the second-order strain gradient theory the constitutive relations under plane stress condition can be expressed as [19]:

$$\begin{cases} \sigma_{xx} = \frac{E}{1-\nu^2}(\epsilon_{xx} + \nu\epsilon_{yy}) - l^2 \frac{E}{1-\nu^2}\nabla^2(\epsilon_{xx} + \nu\epsilon_{yy}) \\ \sigma_{yy} = \frac{E}{1-\nu^2}(\epsilon_{yy} + \nu\epsilon_{xx}) - l^2 \frac{E}{1-\nu^2}\nabla^2(\epsilon_{yy} + \nu\epsilon_{xx}) \\ \tau_{xy} = \frac{E}{2(1+\nu^2)}\gamma_{xy} - l^2 \frac{E}{2(1+\nu^2)}\nabla^2\gamma_{xy} \end{cases} \tag{5}$$

where E is the Young's modulus, ν the Poisson's ratio, l the nonlocal parameter, while $\nabla^2 = \frac{\partial^2(.)}{\partial x^2} + \frac{\partial^2(.)}{\partial y^2}$ denotes the Laplace operator.

The governing equation in terms of out-of-plane deflection w can be derived substituting Eq. (2) and Eqs. (4)-(5) into Eq. (1):

$$\mathcal{R} := D\nabla^4 w - l^2 D\nabla^6 w - q = 0 \quad \text{in } \Omega \tag{6}$$

where $\mathcal{R}$ is the residual function telling how much the equilibrium condition is violated inside the computational domain $\Omega = (0, a) \times (0, b)$, $D = Eh^3/12(1 - \nu^2)$ is the bending rigidity of the nanoplate, while the differential operators $\nabla^4(.)$ and $\nabla^6(.)$ are defined as:

$$\nabla^4(.) = \frac{\partial^4(.)}{\partial x^4} + 2\frac{\partial^4(.)}{\partial x^2 \partial y^2} + \frac{\partial^4(.)}{\partial y^4}$$

$$\nabla^6(.) = \frac{\partial^6(.)}{\partial x^6} + 3\frac{\partial^6(.)}{\partial x^4 \partial y^2} + 3\frac{\partial^6(.)}{\partial x^2 \partial y^4} + \frac{\partial^6(.)}{\partial y^6} \tag{7}$$

Note, the differential order of the PDE rises to six due to the use of a nonlocal theory, as seen in Eq. (6). When the nonlocal parameter l is set to zero, the classical Kirchhoff plate theory is recovered.

To completely define the differential problem in Eq. (6), proper boundary conditions need to be specified. Referring to Figure 1, essential and natural boundary conditions are defined for the four edges i_1, i_2, j_1 and j_2 as:

$$\mathcal{B}^{(j_k)} := \begin{cases} w - \hat{w} = 0 & \text{or} \quad V_x - \hat{V}_x = 0 \\ \frac{\partial w}{\partial x} - \frac{\partial \hat{w}}{\partial x} = 0 & \text{or} \quad M_{xx} - \hat{M}_{xx} = 0 \\ \frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 \hat{w}}{\partial x^2} = 0 & \text{or} \quad H_x - \hat{H}_x = 0 \end{cases} \quad \text{in } \partial\Omega^{(j_k)} \tag{8}$$

and

$$\mathcal{B}^{(i_k)} := \begin{cases} w - \hat{w} = 0 & \text{or} \quad V_y - \hat{V}_y = 0 \\ \frac{\partial w}{\partial y} - \frac{\partial \hat{w}}{\partial y} = 0 & \text{or} \quad M_{yy} - \hat{M}_{yy} = 0 \\ \frac{\partial^2 w}{\partial y^2} - \frac{\partial^2 \hat{w}}{\partial y^2} = 0 & \text{or} \quad H_y - \hat{H}_y = 0 \end{cases} \quad \text{in } \partial\Omega^{(i_k)} \tag{9}$$

where $\mathcal{B}^{(i_k)}$ and $\mathcal{B}^{(j_k)}$ ($k = 1, 2$) are boundary residual functions dependent on the types of conditions enforced on $\partial\Omega^{(i_k)}$ and $\partial\Omega^{(j_k)}$, respectively, while the caret defines any prescribed quantity, either in terms of generalized displacements or forces. The quantities V_x, M_{xx}, H_x are the transverse shear force, bending moment, higher-order bending moment resultants on the edges normal to the axis x, and are defined as follows:

$$V_x = D\left[\frac{\partial^3 w}{\partial x^3} + (2 - \nu)\frac{\partial^3 w}{\partial x \partial y^2}\right] - l^2 D\left[\frac{\partial^5 w}{\partial x^5} + 3\frac{\partial^5 w}{\partial x^3 \partial y^2} + (3 - \nu)\frac{\partial^5 w}{\partial x \partial y^4}\right]$$

$$M_{xx} = -D\left[\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right] + l^2 D\left[\frac{\partial^4 w}{\partial x^4} + \nu\frac{\partial^4 w}{\partial y^4} + (3 - \nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right]$$

$$H_x = l^2 D\left(\frac{\partial^3 w}{\partial x^3} + \nu\frac{\partial^3 w}{\partial x \partial y^2}\right) \tag{10}$$

6

The corresponding generalized forces on the edges normal to the axis y are:

$$V_y = D\left[\frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y}\right] - l^2 D\left[\frac{\partial^5 w}{\partial y^5} + 3\frac{\partial^5 w}{\partial x^2 \partial y^3} + (3-\nu)\frac{\partial^5 w}{\partial x^4 \partial y}\right]$$

$$M_{yy} = -D\left[\nu\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right] + l^2 D\left[\nu\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} + (3-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right]$$

$$H_y = l^2 D\left(\frac{\partial^3 w}{\partial y^3} + \nu\frac{\partial^3 w}{\partial x^2 \partial y}\right) \tag{11}$$

Typical boundary conditions in nanoplate applications can be of Free (F), Simply-supported (S) and Clamped (C) type. These conditions have to take into account both classical and nonlocal boundary conditions, as shown in Table 1. In general there are two possible nonlocal boundary conditions for strain gradient theories [43–45], i.e. imposed curvature or imposed higher-order bending moment resultant. In the present work, the free, simple-support and clamped conditions are selected according to the convention used in [18].

The full problem can be written in a compact form as:

$$\begin{cases} \mathcal{R}(\mathbf{\Lambda}, \mathbf{u}, \mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ \mathcal{B}(\mathbf{\Lambda}, \mathbf{u}, \mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \end{cases} \tag{12}$$

where $\mathcal{B} = \left\{\mathcal{B}^{(i_1)}, \mathcal{B}^{(j_2)}, \mathcal{B}^{(i_2)}, \mathcal{B}^{(j_1)}\right\}^{\mathrm{T}}$ is the boundary residual function collecting all the boundary conditions in $\partial\Omega = \partial\Omega^{(i_1)} \cup \partial\Omega^{(j_2)} \cup \partial\Omega^{(i_2)} \cup \partial\Omega^{(j_1)}$. The vector $\mathbf{\Lambda}$ collects all the parameters of the model, such as the elastic properties, the geometric ones and the nonlocal parameter. Dependently on the problem at hand, all or part of the entries of $\mathbf{\Lambda}$ can be known or values to be identified. For direct problems, all the entries are known; in the inverse case, one or more parameters need to be identified. The vectors $\mathbf{u}$ and $\mathbf{x}$ of Eq. (12) are those collecting the dependent – generalized displacements and/or forces – and independent – spatial positions – variables, respectively.

2.2 Solution via Physics-Informed Neural Networks

The solution of the PDE based on nonlocal strain gradient theory has already been discussed in the context of analytical methods, such as Navier's [19, 20] and Levy's [18] solutions. Numerical approaches have also been proposed in order to consider generic boundary conditions, geometries and materials, see, e.g., Finite Elements [23, 24] and meshless methods [28, 29, 46, 47].

A completely novel strategy for solving the differential problem described by Eq. (12) relies on ANN-based methods, such as Physics-Informed Neural Networks (PINNs) [33]. PINNs have been applied in different fields of science, such as solid mechanics [35]. Recently the authors have proposed their applications for studying composite plate and shell structures at macro scale level [36]. An extension of the work in [36] is presented in this section. Addressed is the use of PINNs for the analysis of plate structures at nanoscale

scale level, where size effects are not negligible.

The convention used for defining Artificial Neural Networks (ANNs) is illustrated in Figure 2. For simplicity the neural networks employed in this work are fixed to have a single-hidden-layer architecture, where N_n is the number of hidden neurons, while $\mathbf{x}$ and $\mathbf{u}$ are the input and output, respectively. The input vector collects the $N_\mathrm{i} = 2$ independent variables of the problem $\mathbf{x} = \{x, y\}^\mathrm{T}$, while the output vector collects the N_o physical quantity to be learnt. These quantities are approximated by the network as follows:

$$\mathbf{u}\left(\mathbf{\Theta}, \mathbf{x}\right) = \mathbf{C}\mathbf{H}\left(\mathbf{W}, \mathbf{b}, \mathbf{x}\right), \quad \text{with} \quad \mathbf{H} = \sigma\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right) \tag{13}$$

where $\mathbf{H}$ is the hidden layer matrix with its generic column vector $\mathbf{h}_p \in \mathbb{R}^{N_\mathrm{n} \times 1}$ representing the output of all hidden neurons due to the p-th input point, σ is the activation function adopted in the hidden layer, while $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{C}\}$ collects all the internal parameters of the networks, i.e. hidden weights $\mathbf{W} \in \mathbb{R}^{N_\mathrm{n} \times N_\mathrm{i}}$, hidden biases $\mathbf{b} \in \mathbb{R}^{N_\mathrm{n} \times 1}$, and output weights $\mathbf{C} \in \mathbb{R}^{N_\mathrm{o} \times N_\mathrm{n}}$.

The internal parameters of the neural network $\mathbf{\Theta}$ are learned through the minimization of a proper loss function $\mathcal{L}$. For PINNs this function is defined as a sum of a data-driven and a physics-guided contributions as:

$$\mathcal{L} = \mathcal{L}_\mathrm{u} + \mathcal{L}_\mathrm{c} \tag{14}$$

The data-driven part can be defined in the form of a mean squared error as:

$$\mathcal{L}_\mathrm{u} = \sum_{k=1}^{N_\mathrm{u}} \frac{|\mathbf{u}_k - \mathbf{u}_k^*|^2}{2N_\mathrm{u}} \tag{15}$$

where $|.|$ is the Euclidean norm, N_u is the total number of available labeled point, $\mathbf{u}_k^* \in \mathbb{R}^{N_\mathrm{o} \times 1}$ is the target value for the k-th input data $\mathbf{x}_k = \{x_k, y_k\}^\mathrm{T}$, while $\mathbf{u}_k = \mathbf{u}\left(\mathbf{\Theta}, \mathbf{x}_k\right) \in \mathbb{R}^{N_\mathrm{o} \times 1}$ is the corresponding neural network prediction.

Similarly, the contribution associated with the mathematical model reads:

$$\mathcal{L}_\mathrm{c} = \sum_{m=1}^{N_\mathrm{f}} \frac{|\mathcal{R}_r - 0_r|^2}{2N_\mathrm{f}} + \sum_{s=1}^{N_\mathrm{b}} \frac{|\mathcal{B}_s - 0_s|^2}{2N_\mathrm{b}} \tag{16}$$

where N_f and N_b are the number of points inside the domain Ω and on its boundaries $\partial\Omega$, respectively, while $\mathcal{R}_r = \mathcal{R}\left(\mathbf{\Lambda}, \mathbf{u}_r, \mathbf{x}_r\right)$ and $\mathcal{B}_s = \mathcal{B}\left(\mathbf{\Lambda}, \mathbf{u}_s, \mathbf{x}_s\right)$ are the residual functions evaluated at the collocation points $N_\mathrm{c} = N_\mathrm{f} + N_\mathrm{b}$.

Different learning algorithm can be employed for minimizing the loss function in Eq. (14) such as Gradient-based learning (GBL) approaches [41] and the Extreme Learning Machine (ELM) [40]. In the present work, the ELM is adopted for its superior computational performance and relatively much easier set up of hyperparameters compared to GBL approaches.

In the context of ELM, the minimization of Eq. (14) is performed in terms of output weights $\mathbf{C}$, while the

other internal parameters $\mathbf{W}$ and $\mathbf{b}$ are chosen randomly and set as constants. The whole training process is condensed into the solution of a linear least-square problem in the form of:

$$\mathbf{L}\mathbf{c} = \mathbf{t} \tag{17}$$

where $\mathbf{c} \in \mathbb{R}^{N_\mathrm{n}N_\mathrm{o}\times 1}$ is the global vector of unknowns collecting all the output weights of the network, while $\mathbf{L} \in \mathbb{R}^{(N_\mathrm{u}+N_\mathrm{f}+N_\mathrm{b})N_\mathrm{o}\times N_\mathrm{n}N_\mathrm{o}}$ and $\mathbf{t} \in \mathbb{R}^{(N_\mathrm{u}+N_\mathrm{f}+N_\mathrm{b})N_\mathrm{o}\times 1}$ are the matrix of coefficients and the vector of targets, respectively. If a single output $N_\mathrm{o} = 1$ is considered, these last two quantities are assembled as:

$$\mathbf{L}\left(\mathbf{\Lambda}, \mathbf{x}\right) = \begin{bmatrix} \mathbf{h}_k^\mathrm{T} \\ \mathcal{R}\left(\mathbf{\Lambda}, \mathbf{h}_r^\mathrm{T}, \mathbf{x}_r\right) \\ \mathcal{B}\left(\mathbf{\Lambda}, \mathbf{h}_s^\mathrm{T}, \mathbf{x}_s\right) \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} u_k^* \\ 0_r \\ 0_s \end{bmatrix} \quad \text{for} \quad \begin{array}{l} k = 1, ..., N_\mathrm{u} \\ r = 1, ..., N_\mathrm{f} \\ s = 1, ..., N_\mathrm{b} \end{array} \tag{18}$$

After a proper parametrization of Eq. (12), the solution of Eq. (17) is sought in a least-square sense through computation of the pseudoinverse of the coefficient matrix $\mathbf{L}$, i.e.:

$$\mathbf{c} = \mathbf{L}^\dagger \mathbf{t} \tag{19}$$

where $\mathbf{L}^\dagger \in \mathbb{R}^{N_\mathrm{n}N_\mathrm{o}\times(N_\mathrm{u}+N_\mathrm{f}+N_\mathrm{b})N_\mathrm{o}}$ is the Moore-Penrose generalized inverse of the rectangular matrix $\mathbf{L}$, and is computed here via Singular Value Decomposition.

Depending on the data types available in Eq. (18), different types of neural networks are obtained. Following the convention introduced in [36], they are defined as follows: black-box ANN ($N_\mathrm{u} > 0$, $N_\mathrm{f} = 0$, $N_\mathrm{b} = 0$), white-box ANN ($N_\mathrm{u} = 0$, $N_\mathrm{f} > 0$, $N_\mathrm{b} > 0$) and grey-box ANN ($N_\mathrm{u} > 0$, $N_\mathrm{f} > 0$, $N_\mathrm{b} > 0$). In this work, the direct solution of PDEs will be performed with PINNs trained using a white-box approach. On the other hand, a grey-box learning approach will be adopted for solving inverse problems.

2.3   Identification via Physics-Informed Neural Networks

A further feature offered by PINNs is the possibility to solve inverse problems, i.e. problems where there is a subset of model parameters $\boldsymbol{\lambda} \subset \boldsymbol{\Lambda}$ which are uncertain, or even unknown, and needs to be identified. The use of PINNs for such applications has been discussed in previous efforts in the literature. For instance, PINNs have been applied for learning unknown model parameters of PDEs in the field of fluid dynamics [33], epidemiology [48] and classical continuum mechanics [49]. However, to the best of the authors' knowledge there are no examples of such applications in the field of micro- and nano-structures.

In the context of PINNs, parameter identification of PDEs is performed starting from a set of observation data $\left\{\mathbf{x}_k, \mathbf{u}_k^*\right\}^\mathrm{T}$. The inverse problem is set up by treating the uncertain model parameters $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda \times 1}$ as unknowns to be tuned together with the output weights $\mathbf{c} \in \mathbb{R}^{(N_\mathrm{n}N_\mathrm{o})\times 1}$. As result, the problem in Eq. (17) turns into a nonlinear least-square problem to be solved:

$$\mathbf{L}\left(\boldsymbol{\beta}\right)\boldsymbol{\beta} = \mathbf{t} \tag{20}$$

where $\boldsymbol{\beta} = \{\boldsymbol{\lambda}, \mathbf{c}\}^{\mathrm{T}} \in \mathbb{R}^{(N_{\mathrm{n}} N_{\mathrm{o}} + N_\lambda)}$ is the vector of global unknowns augmented with the N_λ uncertain parameters, while $\mathbf{L}(\boldsymbol{\beta})$ is the new coefficient matrix which is now nonlinear due to its dependency on all the model parameters, see Eq. (18).

The solution of Eq. (20) is sought in an iterative way by recursively updating the vector of unknowns as:

$$\boldsymbol{\beta}_{(t+1)} = \boldsymbol{\beta}_{(t)} + \boldsymbol{\Delta\beta} \tag{21}$$

where (t) represents the current iteration step, while the updating term $\boldsymbol{\Delta\beta}$ is computed from the solution of the following linear least-square problem:

$$\mathbf{J}_{(t)}\boldsymbol{\Delta\beta} = \mathbf{r}_{(t)} \tag{22}$$

This linear least-square problem is solved at each iteration; the vector of residuals is defined as $\mathbf{r}_{(t)} = \mathbf{L}(\boldsymbol{\beta}_{(t)})\boldsymbol{\beta}_{(t)} - \mathbf{t}$ and the corresponding Jacobian matrix $\mathbf{J}_{(t)} = \frac{\partial \mathbf{r}_{(t)}}{\partial \boldsymbol{\beta}_{(t)}}$. In the present work, the Jacobian is computed exactly using automatic differentiation [50].

The identification process starts from an initial guess $\boldsymbol{\beta}_{(0)} = \left\{\boldsymbol{\lambda}_{(0)}, \mathbf{c}_{(0)}\right\}^{\mathrm{T}}$. From this initial solution, a first evaluation of the residual $\mathbf{r}_{(0)}$ is performed along with its Jacobian $\mathbf{J}_{(0)}$. Then, Eq. (22) is solved to find the updating vector $\boldsymbol{\Delta\beta} = \mathbf{J}_{(0)}^{\dagger}\mathbf{r}_{(0)}$ which is used to generate the new solution $\boldsymbol{\beta}_{(1)} = \boldsymbol{\beta}_{(0)} + \boldsymbol{\Delta\beta}$. The iterative process is continued until either the tolerance criterion $\mathcal{L}_{(t)} < \mathrm{tol}$ or the convergence criterion $|\mathcal{L}_{(t)} - \mathcal{L}_{(t-1)}| < \mathrm{tol}$ are met.

3 Results

The PINN-based framework presented in the previous section is now applied for the bending analysis of Kirchhoff isotropic nanoplates. The code implementing PINNs have been developed in a MATLAB environment, while all the numerical simulations have been performed in a computer with the following characteristics: 1.4 GHz Intel Core i7 processor, 16 GB 1867 MHz LPDDR3 memory, Intel HD Graphics 615 1536 MB.

Goal of this section is demonstrating the potential of PINNs as a numerical technique for solving nonclassical solid mechanics problems with focus on nanoplates. In a first part, an assessment is conducted to validate the code and provide evidence of the correct implementation of PINNs; the second part discusses the role of neural network hyperparameters on the accuracy and stability of the solution; the third part presents a performance assessment of the method through comparison with existing numerical methods; in the last part, the application of PINNs to the solution of an inverse problem is discussed, where goal of the analysis is the identification of the nonlocal parameter.

3.1 Validation

The implementation is validated here against analytical solutions available in the literature. Different combinations of boundary conditions, geometries and values of nonlocal parameter l are considered to verify the

method in wide range of situations.

In the analyses presented next, the nanoplates are characterized by length $a = 10$ nm, variable width b and thickness $h = 0.34$ nm; the material properties are given by $E = 1100$ GPa and $\nu = 0.3$. All these quantities are kept fixed for all the nanoplates considered in the paper, unless otherwise specified. Furthermore, five different sets of boundary conditions are considered. Referring to Figure 1, the conditions are specified following the order $i_1 - j_2 - i_2 - j_1$, and they are given as follows: SSSS, SCSC, SFSF, SCSF and CCCC. The PINN architecture adopted for the validation has $N_{\mathrm{n}} = 1200$ hidden neurons with one output representing the out-of-plane deflection w. The activation function used in the single hidden layer is chosen as the hyperbolic tangent $\sigma(s) = \tanh(s)$. The hidden weights $\mathbf{W}$ and biases $\mathbf{b}$ are randomly initialized in the range $[-2, 2]$ from a uniform Gaussian distribution. The set of training points $\mathbf{x}_p = \{x_p, y_p\}^{\mathrm{T}}$ are normalized in the range $[-1, 1]$ according to the following transformation $\xi = 2x/a - 1$ and $\eta = 2y/b - 1$. These data points are constituted by a grid of $N_{\mathrm{c}} = 30 \times 30$ collocation points distributed in the domain according to the Chebyshev-Gauss-Lobatto distribution:

$$X_c = -\cos\left(\frac{p-1}{N_c - 1}\pi\right) \quad p = 1, 2, ..., N_{\mathrm{c}} \tag{23}$$

A summary of the hyperparameters setting is illustrated in Table 2.

Results are presented in Table 3 in terms of the nondimensional deflection $\alpha = 1000Dw_{\max}/(qa^4)$ for different aspect ratios b/a and values of the nonlocal parameter l. It can be observed that the predictions provided by the PINN are consistent with the analytical results reported in the literature [18]. The hardening effect expected from the negative strain gradient theory is reproduced correctly by the neural network for all combinations of boundary conditions and geometries.

As seen, the error percentage $\mathbb{E}_{\%} = (\alpha - \alpha_{\mathrm{ref}})/\alpha_{\mathrm{ref}} \times 100$ remains bounded well below 0.3%. The maximum error is found to be for the rectangular configuration $b/a = 2$ under the mixed boundary condition CSCS, while in all other cases the errors are of the order of $10^{-3} - 10^{-4}$.

The excellent agreement of these results gives evidence of the correct implementation of the PINN code for different geometries and boundary conditions.

3.2 Parametric studies

Parametric studies on hyperparameters are required to understand the behavior of PINNs when facing a specific problem. Previous works have already addressed these aspects in the context of differential problems in classical mechanics [35, 36]. In this section, similar parametric studies are presented for problems in the context of nonlocal theories where the PDEs are of higher differentiation order, i.e. up to six.

A total of three parametric studies are performed to understand the behavior of PINNs under different training setups. More specifically, the effect of the following hyperparameters will be investigated in this section: collocation points distribution, activation functions and neural network architectures.

11

All the analyses are conducted starting from the baseline network architecture presented in Section 3.1, see Table 2, and considering the rectangular nanoplate of aspect ratio $b/a = 2$ used in the validation phase.

3.2.1 Collocation points distributions

The first study deals with the effect of different distributions of collocation points. Specifically, Chebyshev-Gauss-Lobatto (Che-Gau-Lob), Uniform (Unif) and Random (Rand) distributions are considered, see Figure 3a. The results are compared in terms of maximum ratio between the flexural displacement from nonlocal w^{NL} ($l > 0$) and local theories w^{L} ($l = 0$). These values are reported against the normalized nonlocal parameter $(l/a)^2$ in Figure 3 for different sets of boundary condition. Analytical solutions are used as reference to check the errors. From the figures, one can see that the stiffening behavior shown by the reference solutions [18] is correctly predicted for all distributions of collocation points. However, some local discrepancies can be observed when the nondimensional parameter $(l/a)^2$ assumes values close to zero, i.e. in the limiting region between local and nonlocal theories. In particular, an irregular trend is observed in this range for the Unif and Rand distributions, while the solution is more regular for the Che-Gau-Lob one. This anomalous behavior is very marked in presence of clamped or free edge conditions, see Figures 3c to 3f. We conjecture that the problem stems from the nonclassical boundary conditions on high-order bending moments, i.e. $H_x = 0$ and $H_y = 0$. Indeed, high-order bending moments have a quadratic dependency on the local parameter l, see Eqs. (10) and (11). When l approaches to zero, the rows of $\mathbf{L}$ associated with the natural boundary conditions $H_x = 0$ and/or $H_y = 0$ see a decrease of their Euclidean norm. This causes a reduction of the rank of $\mathbf{L}$ and a consequent poor conditioning of the least-square problem of Eq. (17). This effect is highlighted in Figure 4, where the conditioning number of the coefficient matrix, $k[\mathbf{L}]$, is presented for the three sets of boundary conditions, i.e. SSSS, SCSC and CCCC. As seen, an overall increase of the conditioning of the problem is achieved whenever high-order moments are involved. This effect is particularly marked for values of l close to zero.

By inspection of Figure 4, one can note that the Che-Gau-Lob grid generally guarantees a better conditionality of $\mathbf{L}$. This result is consistent with the findings presented in [36], where a similar study was conducted for macroscopic plates. The considerations above explain the more stable and regular solution observed for the Che-Gau-Lob distribution.

3.2.2 Activation functions

Another parametric study is presented in this section to study the effect of changing the activation function σ in the hidden units. In particular, the plots of Figure 3 are reproduced in Figure 5 using three different expressions for σ, i.e. $\sigma(s) = \tanh(s) \in [-1, 1]$, $\sigma(s) = \mathrm{sigmoid}(s) \in [0, 1]$, $\sigma(s) = \mathrm{atan}(s) \in [-\pi/2, \pi/2]$, while the points distribution is kept fixed as the Che-Gau-Lob, see Table 2.

From Figure 5 one can see that the hardening trend of the nanoplate is captured correctly also when different

activation functions are used. In particular, the PINN generates very similar results when the hyperbolic tangent and arctangent are adopted. This is due to their similarity in shape which is characterized by a s-form bounded between a symmetric interval around zero, see Figure 5a. On the other hand, results with the sigmoid tend to be slightly different and relatively more irregular in some range values of $(l/a)^2$. This behavior is explained by referring to Figure 6a, which shows the $k(\mathbf{L})$ versus $(l/a)^2$ curves given by the different activation functions for the specific case of SSSS condition – similar considerations hold for other boundary conditions. As seen, the adoption of the sigmoid leads to relatively larger conditioning numbers compared to the hyperbolic tangent and arctangent cases. This is due to the more restricted output range of the sigmoid, which falls in $[0, 1]$. By comparison of Figure 5a and Figure 6a one can observe that the output interval of the activation function has a drastic effect on the conditionality of the resulting least-square problem. A wider output interval, as for the arctangent, guarantees better conditioning for the coefficient matrix $\mathbf{L}$.

Despite the relatively higher conditioning numbers, the sigmoid is still able to capture the global solution accurately. This is seen in Figure 6b, where the displacement field and the relative error distribution $\mathbb{E}_{\mathrm{rel}} = |w^{\mathrm{NL}} - w_{\mathrm{ref}}^{\mathrm{NL}}|/w_{\mathrm{ref}}^{\mathrm{L}}$ are plotted for different values of l. For all the cases presented, the relative errors are always below 10^{-2} in all the computational domain. These results give evidence of the reliability of PINNs for the solution of high-order PDEs, even when the training process involves poorly conditioned matrices.

3.2.3 Network architectures

In this section, the use of different network architectures is explored and their effectiveness in the solution process is discussed. Three different architectures, here referred as NET1, NET2 and NET3, are proposed and compared to each other. These network setups differ in the number of hidden neurons N_{n} and outputs N_{o}, while they share all the other hyperparameters, see Table 2.

The architecture NET1 is recovered from the PINN used in Section 1.1. It has $N_{\mathrm{n}} = 1200$ hidden neurons and $N_{\mathrm{o}} = 1$ outputs given by $\mathbf{u} = \{w\}^{\mathrm{T}}$. The configuration of NET2 has a number of neuron equal to $N_{\mathrm{n}} = 2400$, which is the double with respect to those of NET1. The third network, NET3, is characterized by the output vector $\mathbf{u} = \{w, M_{xx}, M_{xy}, M_{yy}\}^{\mathrm{T}}$, while its hidden layer has the same height of NET1. For clarity, a graphical representation of the different architectures is presented in Figure 7a.

Owing to the different choice of network outputs, the residual functions and the boundary conditions have to be appropriately specified, as per Table 4. A purely displacement-based formulation is employed for NET1 and NET2, where the output is given by the deflection w. A displacement-stress-based formulation is obtained in the case of NET3, where the output vector $\mathbf{u}$ includes both the deflection and some stress-related quantities. For this last architecture, the problem formulation has been modified to have also the three bending moments as field variables. More specifically, the constitutive relations have been added

13

in the set of governing equations and the equilibrium condition has been rewritten in terms of moment resultants. Alternative approaches to formulate strain gradient problems in terms of kinetic variables exist in the literature. Some examples can be found in [51], where the Airy stress function is used as primary unknown variable, and in [52], where both kinematic and kinetic fields variables are used to formulate the elastic problem.

The ratio between the maximum deflections obtained with local and nonlocal theories, $w^{\mathrm{NL}}/w^{\mathrm{L}}$, is plotted in Figures 7b to 7f against the nondimensional parameter $(l/a)^2$. The curves refer to the three different PINN architectures presented earlier. The results for NET1 and NET2 are exactly the same, meaning that a number of neurons $N_{\mathrm{n}} = 1200$ guarantees convergence. The additional neurons of NET2 do not provide any beneficial effects on the solution.

Looking at Figures 7d and 7e, one can observe that the predictions provided by NET3 are less accurate with respect to the other networks. This behaviour stems from the additional conditions – the three constitutive relations – imposed in the plate domain. In particular, the constitutive relations represent extra information which NET3 has to learn. This makes the training process harder to be carried out compared with the cases of NET1 and NET2, especially if the same number of neurons is adopted.

In absence of free edge conditions, NET3 provides similar results compared to the other two architectures, see Figures 7b to 7c and Figure 7f. This can be explained looking at the differential order of the governing equations. Indeed, a displacement-based formulation leads to a differential problem of order six, irrespectively on the boundary conditions. On the contrary, the order using a displacement-stress-based formulation is equal to five in the presence of at least one free edge, and four in all other cases. As it turns out, NET3 is trained with differential equations of two orders lower with respect to NET1 and NET2 when the three boundary conditions SSSS, SCSC and CCCC, are considered. This enable to counterbalance the higher number of equations to be handled by NET3.

The comparison in terms of displacements does not reveal any advantage in using NET3. However, the benefit of a displacement-stress-based formulation can be fully appreciated by inspection of force-related quantities, see Figure 8, where the moment resultants M_{xx} and M_{yy} are plotted versus $(l/a)^2$ for SCSC boundary conditions. Here NET3 provides more regular trends and closer agreement to the reference solutions with respect to NET1 and NET2. Indeed, the bending moments are directly available as an output of the network and do not require any derivative to be computed in the post-processing phase.

The results of this parametric study illustrate that the definition of an optimal network architecture is problem-dependent, and has to be found on the basis of the boundary conditions and the results of interest, e.g. displacements or stresses. In the context of the proposed framework, switching from one architecture to another is a relatively simple task, so the method offers a great degree of flexibility in handling different kind of plate problems.

14

3.3 Performance evaluation

A comparison in terms of accuracy and efficiency with state-of-the-art numerical methods is presented in this section. The performance of our PINN method is assessed against two different FE-based approaches to solve strain gradient nanoplates available in the literature. The first FE method has a formulation based on conforming elements (FEM1), while the second one on non-conforming elements (FEM2). In the following study, the solutions presented for FEM1 and FEM2 are referred to a numerical model with a mesh size of 32×32 [23].

The nanoplate considered for this study has the same characteristics of the ones presented in Section 3.1. The aspect ratio is fixed to $b/a = 1$, while the boundary conditions are of type SSSS. The network architecture and hyperparameters are chosen based on the parameter studies conducted in Section 3.1, and are summarized in Table 2.

Results are presented in Table 5 in terms of computational times and percentage errors for different values of the length scale parameter. It can be observed that among the methods the PINN approach achieves the highest accuracy at the lowest computational cost. More specifically, the computational time of PINNs is almost haft of the one of FEM1 and one fourth of FEM2.

The superior effectiveness of PINNs relies on two main aspects. One is their meshfree nature which eliminates any sort of domain discretization for the approximation of the unknown fields. Differently from FE-based approaches, the enforcement of inter-element continuity of displacements and their derivatives is not required here. This gives a greater flexibility when one wants to refine the solution. In the present framework, solution refinement is performed by simply increasing the number of neurons and collocation points.

The other aspect regards the adoption of the ELM which permits to have a much faster learning process compared with standard GBL approaches. This allows ML practitioners to finally break the computational barrier set by the training process which hinders many existing PINN-based methods [37–39]. In this regard, a drastic saving in computational times can be achieved thanks to ELM, as training is accomplished in a single step through solution of the least-square problem in Eq. (17).

This study provides clear evidence of the potentiality offered by PINNs and the ELM for the direct solution of PDEs. For example, the improved efficiency of the PINN-ELM framework can be exploited to perform parametric analysis to study the effect of geometric and mechanical parameters on the nanoplate response. This can be performed without the need of ML concepts such as Transfer Learning (TL), typically employed in PINN-GBL approaches to accelerate training [53]. Moreover, our framework is so flexible that can be generalized to any mathematical model, ranging from CCTs to NCTs, as well as from nanobeam models to nanoplate/nanoshell models.

3.4 Parameters identification

This section discusses the application of PINNs for solving inverse problems. The structure under investigation is a nanoplate loaded with a uniform pressure q and constrained at the four edges with simply-supported conditions (SSSS). The known parameters of the mathematical model are the dimensions of the plate, $a = 10$ nm, $b = 20$ nm, $h = 0.34$ nm, and the elastic properties of the material, $E = 1100$ MPa, $\nu = 0.3$. The identification of the nonlocal parameter l is discussed starting from the knowledge of its governing equations, Eq. (12), along with available data measurements $\left\{\mathbf{x}_p, \mathbf{u}_p^*\right\}^{\mathrm{T}}$. In this study, these measurements are artificially generated in a number of points by solving the same problem with a white-box approach and fixing $l = 1$ nm. This value of the length scale parameter has been arbitrary chosen for the inverse problem, however in practice its value is determined from experimental studies.

The identification is performed by means of a PINN with a single-hidden-layer architecture with $N_{\mathrm{n}} = 1000$ and one single output $\mathbf{u} = \{w\}^{\mathrm{T}}$. The set of training points is built with a uniform grid of $N_{\mathrm{c}} = 30 \times 30$ collocation points and $N_{\mathrm{u}} = 3000$ artificial measurements randomly distributed in the domain, see Table 2. A random noise $\left\{\mathbf{x}_k, w_k^* + n_k\right\}^{\mathrm{T}}$ ($k = 1, ..., N_u$) is superposed to the measures in order to simulate more realistic working conditions. In order to test the robustness of PINN, the inverse problem is solved by considering: different initial guesses for the unknown parameter $l_{(0)}$; different levels of noise, quantified as: $n_\% = \max\left(|w_k^* - n_k|\right)/\max\left(w_k^*\right) \times 100$.

In Figure 9 the evolution of the training process is illustrated in terms of loss function ratio $\mathcal{L}_{(t)}/\mathcal{L}_{(0)}$ and error fraction $l_{(t)}/l_{\mathrm{true}}$. In absence of noise, $n_\% = 0$, the PINN correctly identifies the unknown parameter l with an error percentage of the order of $1\% - 2\%$. A summary of the results is available in Table 6, where the training outcome is reported in terms of loss function, fraction error and iterations. As seen, the training process requires few iterations to reach convergence when the initial guess is close the actual value. For instance, just 2 iterations are required when $l_{(0)}/l_{\mathrm{true}} = 0.9$. The number of iterations tends to increase as the ratio $l_{(0)}/l_{\mathrm{true}}$ gets larger. Examples are given by $l_{(0)}/l_{\mathrm{true}} = 1.5$ and $l_{(0)}/l_{\mathrm{true}} = 3.0$, where convergence is reached after 3 and 5 iterations, respectively. In other cases, the training process gets stacked in local minima, see the results for $l_{(t)}/l_{\mathrm{true}} = 0.3$, while no convergence is achieved when $l_{(t)}/l_{\mathrm{true}} = 0.1$. It is therefore necessary to repeat the identification process with different initializations of the unknown parameter in order to be confident with the solution found. This aspect represents clearly a limitation of the present machine learning approach when solving inverse problems. Despite this, the correct solution can generally be found unless the initial guess is too far away from the actual value.

A general worsening of the identified parameter is observed when the measures are affected by noise, see Figures 9b to 9c and Figures 9e to 9f. Starting from an initialization $l_{(0)}/l_{\mathrm{true}} = 1.5$, the loss function ratio at convergence increases of one order of magnitude for $n_\% = 1\%$, and of two orders for $n_\% = 3\%$. The introduction of noise also affects the number of iterations to reach convergence, as clear from Table 6. These results demonstrate the importance of the quality of measures in the identification process. Poor

measures may lead to poor parameter identification. However, a certain degree of robustness is exhibited by PINNs, which lead to relatively accurate identifications, i.e. errors below 2%, for noise amplitudes equal to $n_\% = 1\%$.

The identification approach presented above can be generalized to any nanoplate models with an arbitrary number of independent length scale parameters. The optimal value of these parameters, for different materials/geometries/boundary conditions, can be identified by a PINN trained with the equations of the mathematical model and the solution data N_u. Despite in this study solution data were artificially generated from the same mathematical model, in real situations they can represent experimental measurements and/or results coming from molecular mechanics simulations.

4 Conclusions

The present paper has shown an application of PINNs for the bending analysis of thin nanoplates. PINNs are a new family of ANNs which can incorporate the PDEs of a mathematical model in their training process and some data coming from the physical process under investigation. The result is a hybrid approach which puts together features of conventional numerical methods and ML techniques.

Applications of this new framework are illustrated for two different problems. The first one deals with the direct solution of the nanoplate bending problem. In this regard, the governing equations are derived within the context of the second-order negative strain gradient theory. In the second application, PINNs are used to solve an inverse problem where the nonlocal parameter l is uncertain. The identification is carried out relying upon some data bearing information on the flexural response of the nanoplate.

In both example of applications, PINNs demonstrates to be an effective and robust approach. In particular, for the direct problem the stability and reliability of the method was tested considering different combinations of hyperparameters and networks architectures. It was found that a certain degree of tuning is required to have a properly trained network. This makes this approach less user-friendly if compared with other methods, such as finite elements. However, we have also demonstrated that once a suitable network setup is found our approach can outperform well-established numerical methods in terms of accuracy and efficiency.

Regarding the inverse problem, the sensibility of PINNs to data noise and solution initialization was assessed. It was shown that the quality of training data and the choice of initial guess for the unknown model parameters have a certain influence on the training outcomes. In spite of this, the present method is still capable of performing successfully the identification task, even if data are noisy and the starting guess of unknown parameter is very different from the actual one. This is possible thanks to the exploitation of the physics information of the problem which plays a regularization effect whenever the available knowledge is limited or even partially incorrect.

17

We are confident that the powerful combination of PINNs and ELM can be exploited in future for applications such as structural analysis, design, as well as model identification of more complex nano-structural members, such as laminated nanoplates and nanoshells. Future works will address these aspects.

References

[1] Power AC, Gorey B, Chandra S, and Chapman J. Carbon nanomaterials and their application to electrochemical sensors: a review. Nanotechnology Reviews 2018, 7(1):19–41, https://doi.org/10.1515/ntrev-2017-0160.

[2] Niu C, Sichel EK, Hoch R, Moy D, and Tennent H. High power electrochemical capacitors based on carbon nanotube electrodes. Applied Physics Letters 1997, 70(11):1480–1482, https://doi.org/10.1063/1.118568.

[3] Rosen R, Simendinger W, Debbault C, Shimoda H, Fleming L, Stoner B, and Zhou O. Application of carbon nanotubes as electrodes in gas discharge tubes. Applied Physics Letters 2000, 76(13):1668–1670, https://doi.org/10.1063/1.126130.

[4] Dai H, Hafner JH, Rinzler AG, Colbert DT, and Smalley RE. Nanotubes as nanoprobes in scanning probe microscopy. Nature 1996, 384(6605):147, https://doi.org/10.1038/384147a0.

[5] Chandel VS, Guannan W, and Mohammad T. Advances in modelling and analysis of nano structures: a review. Nanotechnology Reviews 2020, 9(1):230–258, https://doi.org/10.1515/ntrev-2020-0020.

[6] Stölken JS and Evans AG. A microbend test method for measuring the plasticitylength scale. Acta Materialia 1998, 46(14):5109–5115, https://doi.org/10.1016/S1359-6454(98)00153-0.

[7] Lam DCC, Yang F, Chong ACM, Wang J, and Tong P. Experiments and theory in strain gradient elasticity. Journal of the Mechanics and Physics of Solids 2003, 51(8):1477–1508, https://doi.org/10.1016/S0022-5096(03)00053-X.

[8] McFarland AW and Colton JS. Role of material microstructure in plate stiffness with relevance to microcantilever sensors. Journal of Micromechanics and Microengineering 2005, 15(5):1060, https://doi.org/10.1088/0960-1317/15/5/024.

[9] Eringen AC. On differential equations of nonlocal elasticity and solutions of screw dislocation and surface waves. Journal of Applied Physics 1983, 54(9):4703–4710, https://doi.org/10.1063/1.332803.

[10] Aifantis EC. On the role of gradients in the localization of deformation and fracture. International Journal of Engineering Science 1992, 30(10):1279–1299, https://doi.org/10.1016/0020-7225(92)90141-3.

[11] Mindlin RD and Tiersten HF. Effects of couple-stresses in linear elasticity. Archive for Rational Mechanics and Analysis 1962, 11:415–448.

[12] Di Paola M, Failla G, Pirrotta A, Sofi A, and Zingales M. The mechanically based non-local elasticity: an overview of main results and future challenges. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 1993, 371:20120433, https://doi.org/10.1098/rsta.2012.0433.

[13] Yang F, Chong ACM, Lam DCC, and Tong P. Couple stress based strain gradient theory for elasticity. International Journal of Solids and Structures 2002, 39(10):2731–2743, https://doi.org/10.1016/S0020-7683(02)00152-X.

[14] Aifantis EC. Update on a class of gradient theories. Mechanics of Materials 2003, 35(3-6):259–280, https://doi.org/10.1016/S0167-6636(02)00278-8.

[15] Faghidian SA. Flexure mechanics of nonlocal modified gradient nano-beams. Journal of Computational Design and Engineering 2021, 8(3):949–959, https://doi.org/10.1093/jcde/qwab027.

[16] Faghidian SA. Two-phase local/nonlocal gradient mechanics of elastic torsion. Mathematical Methods in the Applied Sciences 2020, https://doi.org/10.1002/mma.6877.

[17] Faghidian SA, Żur KK, Reddy JN, and Ferreira AJM. On the wave dispersion in functionally graded porous timoshenko-ehrenfest nanobeams based on the higher-order nonlocal gradient elasticity. Composite Structures 2022, 279:114819, https://doi.org/10.1016/j.compstruct.2021.114819.

[18] Babu B and PateL BP. Analytical solution for strain gradient elastic Kirchhoff rectangular plates under transverse static loading. European Journal of Mechanics-A/Solids 2019, 73:101–111, https://doi.org/10.1016/j.euromechsol.2018.07.007.

[19] Papargyry-Beskou S and Beskos DE. Static, stability and dynamic analysis of gradient elastic flexural Kirchhoff plates. Archive of Applied Mechanics 2008, 78(8):625–635, https://doi.org/10.1007/s00419-007-0166-5.

[20] Cornacchia F, Fantuzzi N, Luciano R, and Penna R. Solution for cross-and angle-ply laminated Kirchhoff nano plates in bending using strain gradient theory. Composites Part B: Engineering 2019, 173:107006, https://doi.org/10.1016/j.compositesb.2019.107006.

[21] Akgöz B and Ömer C. A microstructure-dependent sinusoidal plate model based on the strain gradient elasticity theory. Acta mechanica 2015, 226(7):2277–2294, https://doi.org/10.1007/s00707-015-1308-4.

[22] Dastjerdi S and Akgöz B. New static and dynamic analyses of macro and nano fgm plates using exact three-dimensional elasticity in thermal environment. Composite Structures 2018, 192:626–641, https://doi.org/10.1016/j.compstruct.2018.03.058.

[23] Babu B and Patel BP. A new computationally efficient finite element formulation for nanoplates using second-order strain gradient Kirchhoff's plate theory. Composites Part B: Engineering 2019, 168:302–311, https://doi.org/10.1016/j.compositesb.2018.12.066.

[24] Bacciocchi M, Fantuzzi N, and Ferreira AJM. Conforming and nonconforming laminated finite element Kirchhoff nanoplates in bending using strain gradient theory. Computers & Structures 2020, 239:106322, https://doi.org/10.1016/j.compstruc.2020.106322.

[25] Niiranen J, Kiendl J, Niemi AH, and Reali A. Isogeometric analysis for sixth-order boundary value problems of gradient-elastic Kirchhoff plates. Computer Methods in Applied Mechanics and Engineering 2017, 316:328–348, https://doi.org/10.1016/j.cma.2016.07.008.

[26] Zhang B, Li H, Kong L, Zhang X, and Shen H. Strain gradient differential quadrature kirchhoff plate finite element with the c2 partial compatibility. European Journal of Mechanics-A/Solids 2020, 80:103879, https://doi.org/10.1016/j.euromechsol.2019.103879.

[27] Liu GR and Gu YT. An introduction to meshfree methods and their programming. Springer Science & Business Media, Dordrecht, Netherlands, 2005.

[28] Fabbrocino F, Saitta S, Vescovini R, Fantuzzi N, and Luciano R. Meshless computational strategy for higher order strain gradient plate models. Mathematical and Computational Applications 2022, 27(2):19, https://doi.org/10.3390/mca27020019.

[29] Wang B, Lu C, Fan C, and Zhao M. A stable and efficient meshfree galerkin method with consistent integration schemes for strain gradient thin beams and plates. Thin-Walled Structures 2020, 153:106791, https://doi.org/10.1016/j.tws.2020.106791.

[30] Wang B, Lu C, Fan C, and Zhao M. A meshfree method with gradient smoothing for free vibration and buckling analysis of a strain gradient thin plate. Engineering Analysis with Boundary Elements 2021, 132:159–167, https://doi.org/10.1016/j.enganabound.2021.07.014.

[31] Roque CMC and Żur KK. On the static, vibration, and transient responses of micro-plates made of materials with different microstructures. Engineering Analysis with Boundary Elements 2022, 143:579–590, https://doi.org/10.1016/j.enganabound.2022.07.007.

[32] Khorasani VS and Bayat M. Bending analysis of fg plates using a general third-order plate theory with modified couple stress effect and mlpg method. Engineering Analysis with Boundary Elements 2018, 94:159–171, https://doi.org/10.1016/j.enganabound.2018.06.015.

[33] Raissi M, Perdikaris P, and Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 2019, 378:686–707, https://doi.org/10.1016/j.jcp.2018.10.045.

[34] Zhang D, Lu L, Guo L, and Karniadakis GE. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. Journal of Computational Physics 2019, 397:108850, https://doi.org/10.1016/j.jcp.2019.07.048.

[35] Haghighat E, Raissi M, Moure A, Gomez H, and Juanes R. A deep learning framework for solution and discovery in solid mechanics. arXiv preprint 2020, 2003.02751, https://doi.org/10.48550/arXiv.2003.02751.

[36] Yan CA, Vescovini R, and Dozio L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. Computers & Structures 2022, 265:106761, https://doi.org/10.1016/j.compstruc.2022.106761.

[37] Guo H, Zhuang X, and Rabczuk T. A deep collocation method for the bending analysis of kirchhoff plate. CMC-Computers, Materials and Continua 2019, 59(2):433–456, https://doi.org/10.32604/cmc.2019.06660.

[38] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, Zhuang X, and Rabczuka T. 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 2020, 362:112790, https://doi.org/10.1016/j.cma.2019.112790.

[39] Zhuang X, Guo H, Alajlan N, Zhu H, and Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of kirchhoff plates with transfer learning. European Journal of Mechanics / A Solids 2021, 87:104225, https://doi.org/10.1016/j.euromechsol.2021.104225.

[40] Huang GB, Zhu QY, and Siew CK. Extreme learning machine: theory and applications. Neurocomputing 2006, 70(1-3):489–501, https://doi.org/10.1016/j.neucom.2005.12.126.

[41] Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint 2016, 1609.04747, https://doi.org/10.48550/arXiv.1609.04747.

[42] Reddy JN. Mechanics of Laminated Composite Plates and Shells: Theory and Analysis. CRC Press, Boca Raton, 2004.

[43] Akgöz B and Ömer C. Analysis of micro-sized beams for various boundary conditions based on the strain gradient elasticity theory. Archive of Applied Mechanics 2012, 82(3):423–443, https://doi.org/10.1007/s00419-011-0565-5.

[44] Akgöz B and Ömer C. Buckling analysis of functionally graded microbeams based on the strain gradient theory. Acta Mechanica 2013, 224(9):2185–2201, https://doi.org/10.1007/s00707-013-0883-5.

[45] Kahrobaiyan MH, Asghari M, and Ahmadian MT. Strain gradient beam element. Finite Elements in Analysis and Design 2013, 68:63–75, https://doi.org/10.1016/j.finel.2012.12.006.

[46] Saitta S, Luciano R, Vescovini R, and Fantuzzi N. Optimization of a radial point interpolation meshless strategy for strain gradient nanoplates. Engineering Analysis with Boundary Elements 2022, 140:70–78, https://doi.org/10.1016/j.enganabound.2022.03.026.

[47] Saitta S, Fabbrocino F, Vescovini R, Fantuzzi N, and Luciano R. Radial point interpolation method for isotropic nanoplates in bending using strain gradient theory. International Journal of Computational Methods 2022, pages 1–22, https://dx.doi.org/10.1142/S0219876222500232.

[48] Schiassi E, D'Ambrosio A, De Florio M, Furfaro R, and Curti F. Physics-informed extreme theory of functional connections applied to data-driven parameters discovery of epidemiological compartmental models. arXiv preprint 2020, 2008.05554, https://doi.org/10.48550/arXiv.2008.05554.

[49] Haghighat E, Raissi M, Moure A, Gomez H, and Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Computer Methods in Applied Mechanics and Engineering 2021, 379:113741, https://doi.org/10.1016/j.cma.2021.113741.

[50] Baydin AG, Pearlmutter BA, Radul AA, and Siskind JM. Automatic differentiation in machine learning: a survey. Journal of Marchine Learning Research 2018, 18(153):1–43, https://www.jmlr.org/papers/volume18/17-468/17-468.pdf.

[51] Farrahi GH, Faghidian SA, and Smith DJ. An inverse approach to determination of residual stresses induced by shot peening in round bars. International Journal of Mechanical Sciences 2009, 51(9-10):726–731, https://doi.org/10.1016/j.ijmecsci.2009.08.004.

[52] Faghidian SA, Żur KK, Pan E, and Kim J. On the analytical and meshless numerical approaches to mixture stress gradient functionally graded nano-bar in tension. Engineering Analysis with Boundary Elements 2022, 134:571–580, https://doi.org/10.1016/j.enganabound.2021.11.010.

[53] Goswami S, Anitescu C, Chakraborty S, and Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theoretical and Applied Fracture Mechanics 2020, 106(102447), https://doi.org/10.1016/j.tafmec.2019.102447.

Table 1: Boundary conditions.

	Edges i		Edges j	
	Classical	Non-classical	Classical	Non-classical
F	$V_y = 0$		$V_x = 0$	
	$M_y = 0$		$M_x = 0$	
		$H_y = 0$		$H_x = 0$
S	$w = 0$		$w = 0$	
	$M_y = 0$		$M_x = 0$	
		$\frac{\partial^2 w}{\partial y^2} = 0$		$\frac{\partial^2 w}{\partial x^2} = 0$
C	$w = 0$		$w = 0$	
	$\frac{\partial w}{\partial y} = 0$		$\frac{\partial w}{\partial x} = 0$	
		$H_y = 0$		$H_x = 0$

Table 2: Hyperparameters settings for the different problems: validation (P1), parametric studies on collocation points distributions (P2.1), activation functions (P2.2), network architectures (P2.3), performance evaluation (P3), and identification (P4).

Problem	Network architecture		Activation	Data number	Data distribution	
	$[N_i/N_n/N_o]$	Output	$\sigma(s)$	$[N_u/N_c]$	Solution	Collocation
P1	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
P2.1	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Unif
	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Rand
P2.2	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
	$[2/1200/1]$	$\{w\}^T$	atan	$[0/900]$	-	Che-Gau-Lob
	$[2/1200/1]$	$\{w\}^T$	sigmoid	$[0/900]$	-	Che-Gau-Lob
P2.3	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
	$[2/2400/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
	$[2/1200/1]$	$\{w, M_{xx}, M_{yy}, M_{xy}\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
P3	$[2/1200/1]$	$\{w\}^T$	tanh	$[0/900]$	-	Che-Gau-Lob
P4	$[2/1000/1]$	$\{w\}^T$	tanh	$[3000/900]$	Rand	Unif

Table 3: Nondimensional deflection, $\alpha = 1000Dw_{\max}/(qa^4)$, under uniform distributed pressure q for a rectangular Kirchhoff nanoplate with different boundary conditions and aspect ratios b/a.

	l(nm)	$b/a = 1$			$b/a = 2$			$b/a = 3$		
		PINN	Analytical	$\mathbb{E}_\%$	PINN	Analytical	$\mathbb{E}_\%$	PINN	Analytical	$\mathbb{E}_\%$
SSSS	0.0	4.0624	4.0624	-0.0012	10.1287	10.1287	-0.0004	12.2328	12.2328	0.0002
	0.2	4.0331	4.0330	0.0014	10.0829	10.0833	-0.0036	12.1797	12.1839	-0.0343
	0.5	3.8844	3.8844	0.0006	9.8502	9.8502	0.0004	11.9323	11.9324	-0.0008
	1.0	3.4231	3.4231	0.0003	9.0886	9.0886	-0.0001	11.1046	11.1046	0.0004
SCSC	0.0	1.9171	1.9171	0.0021	8.4450	8.4450	0.0002	11.6813	11.6813	0.0001
	0.2	1.8796	1.8783	0.0700	8.4128	8.3887	0.2871	11.6364	11.6286	0.0673
	0.5	1.7093	1.7093	0.0009	8.1145	8.1145	-0.0005	11.3614	11.3623	-0.0083
	1.0	1.3040	1.3040	-0.0035	7.2820	7.2820	0.0003	10.5039	10.5039	0.0000
SFSF	0.0	15.0113	15.0113	-0.0003	15.2022	15.2022	-0.0001	15.2180	15.2181	-0.0003
	0.2	14.9477	14.9470	0.0048	15.1420	15.1373	0.0314	15.1580	15.1532	0.0314
	0.5	14.6165	14.6165	-0.0002	14.8041	14.8040	0.0006	14.8197	14.8196	0.0009
	1.0	13.5415	13.5415	-0.0003	13.7212	13.7211	0.0005	13.7361	13.7361	0.0001
SCSF	0.0	11.2359	11.2359	0.0004	14.9491	14.9491	0.0000	15.2034	15.2035	-0.0004
	0.2	11.1716	11.1703	0.0121	14.8903	14.8838	0.0439	15.1433	15.1385	0.0320
	0.5	10.8453	10.8454	-0.0006	14.5494	14.5493	0.0004	14.8046	14.8049	-0.0022
	1.0	9.8416	9.8416	-0.0002	13.4658	13.4658	-0.0001	13.7212	13.7212	-0.0001
CCCC	0.0	1.2653	1.2653	0.0015	2.5330	2.5330	-0.0016	2.6172	2.6172	0.0013
	0.2	1.2351	1.2333	0.1480	2.4862	2.4861	0.0023	2.5713	2.5706	0.0286
	0.5	1.0979	1.0979	0.0020	2.2799	2.2799	0.0012	2.3634	2.3639	-0.0208
	1.0	0.7946	0.7946	0.0057	1.7741	1.7741	-0.0017	1.8518	1.8518	-0.0017

Table 4: Mathematical model definition for different formulations.

	Displacement formulation: $\mathbf{u} = \{w\}^{\mathrm{T}}$	Displacement-stress formulation: $\mathbf{u} = \{w, M_{xx} M_{xy}, M_{yy}\}^{\mathrm{T}}$
$\mathcal{R}$:	Equilibrium condition: $$D\nabla^4 w - l^2 D \nabla^6 w - q = 0$$	Equilibrium condition: $$\frac{\partial^2 M_{xx}}{\partial x^2} + 2\frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_{yy}}{\partial y^2} + q = 0$$ Constitutive relations: $$M_{xx} + D\left[\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right] +$$ $$-l^2 D\left[\frac{\partial^4 w}{\partial x^4} + \nu\frac{\partial^4 w}{\partial y^4} + (3-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right] = 0$$ $$M_{yy} + D\left[\nu\frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial x^2}\right] +$$ $$-l^2 D\left[\nu\frac{\partial^4 w}{\partial y^4} + \frac{\partial^4 w}{\partial x^4} + (3-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right] = 0$$ $$M_{xy} + (1-\nu) D\frac{\partial^2 w}{\partial x \partial y} +$$ $$-l^2 (1-\nu) D\left(\frac{\partial^4 w}{\partial x^3 \partial y} + \frac{\partial^4 w}{\partial x \partial y^3}\right) = 0$$
$\mathcal{B}$:	Condition on transverse shear (V_x, V_y): $$D\left[\frac{\partial^3 w}{\partial x^3} + (2-\nu)\frac{\partial^3 w}{\partial x \partial y^2}\right] +$$ $$-l^2 D\left[\frac{\partial^5 w}{\partial x^5} + 3\frac{\partial^5 w}{\partial x^3 \partial y^2} + (3-\nu)\frac{\partial^5 w}{\partial x \partial y^4}\right] = 0$$ $$D\left[\frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y}\right] +$$ $$-l^2 D\left[\frac{\partial^5 w}{\partial y^5} + 3\frac{\partial^5 w}{\partial x^2 \partial y^3} + (3-\nu)\frac{\partial^5 w}{\partial x^4 \partial y}\right] = 0$$ Condition on bending moments (M_{xx}, M_{yy}): $$-D\left[\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right] +$$ $$+l^2 D\left[\frac{\partial^4 w}{\partial x^4} + \nu\frac{\partial^4 w}{\partial y^4} + (3-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right] = 0$$ $$-D\left[\nu\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right] +$$ $$+l^2 D\left[\nu\frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial y^4} + (3-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2}\right] = 0$$ Condition on high-order bending moments (H_x, H_y): $$l^2 D\left(\frac{\partial^3 w}{\partial x^3} + \nu\frac{\partial^3 w}{\partial x \partial y^2}\right) = 0$$ $$l^2 D\left(\frac{\partial^3 w}{\partial y^3} + \nu\frac{\partial^3 w}{\partial x^2 \partial y}\right) = 0$$	Condition on transverse shear (V_x, V_y): $$D\left[\frac{\partial^3 w}{\partial x^3} + (2-\nu)\frac{\partial^3 w}{\partial x \partial y^2}\right] +$$ $$-l^2 D\left[\frac{\partial^5 w}{\partial x^5} + 3\frac{\partial^5 w}{\partial x^3 \partial y^2} + (3-\nu)\frac{\partial^5 w}{\partial x \partial y^4}\right] = 0$$ $$D\left[\frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y}\right] +$$ $$-l^2 D\left[\frac{\partial^5 w}{\partial y^5} + 3\frac{\partial^5 w}{\partial x^2 \partial y^3} + (3-\nu)\frac{\partial^5 w}{\partial x^4 \partial y}\right] = 0$$ Condition on bending moments (M_{xx}, M_{yy}): $$M_{xx} = 0$$ $$M_{xx} = 0$$ Condition on high-order bending moments (H_x, H_y): $$l^2 D\left(\frac{\partial^3 w}{\partial x^3} + \nu\frac{\partial^3 w}{\partial x \partial y^2}\right) = 0$$ $$l^2 D\left(\frac{\partial^3 w}{\partial y^3} + \nu\frac{\partial^3 w}{\partial x^2 \partial y}\right) = 0$$

Table 5: Comparison between PINN, FEM with nonconforming and nonconforming elements for a rectangular Kirchhoff nanoplate with SSSS boundary conditions and aspect ratios $b/a = 1$.

	l (nm)	Analytical	PINN	FEM1	FEM2
α ($\mathbb{E}_\%$)	0.0	4.0624 (-)	4.0624 (-0.0012)	4.0624 (0.0000)	4.0624 (0.0000)
	0.2	4.0330 (-)	4.0331 (0.0014)	4.0332 (0.0050)	4.0331 (0.0025)
	0.5	3.8844 (-)	3.8844 (0.0006)	3.8856 (0.0309)	3.8845 (0.0026)
	1.0	3.4231 (-)	3.4231 (0.0003)	3.4271 (0.1169)	3.4233 (0.0058)
CPU (s)			12.6606	23.2037	49.2881

Table 6: Identification of nonlocal parameter l.

$l_{(0)}/l_{\mathrm{true}}$	$n_\% = 0\%$			$n_\% = 1\%$			$n_\% = 3\%$		
	$\mathcal{L}_{(t)}$	$l_{(t)}/l_{\mathrm{true}}$	t	$\mathcal{L}_{(t)}$	$l_{(t)}/l_{\mathrm{true}}$	t	$\mathcal{L}_{(t)}$	$l_{(t)}/l_{\mathrm{true}}$	t
0.1	-	-	-	-	-	-	-	-	-
0.3	8.71e-05	0.7693	6	12.83e-05	0.7215	6	25.64e-05	0.6868	7
0.9	1.33e-05	1.0157	2	6.37e-05	0.9787	3	18.79e-05	0.8618	4
1.5	0.88e-05	1.0176	3	6.19e-05	0.9805	4	18.67e-05	0.8680	5
3.0	0.66e-05	1.0180	5	6.08e-05	0.9830	5	17.63e-05	0.8642	6

Figure 1: Nanoplate geometry and reference system.
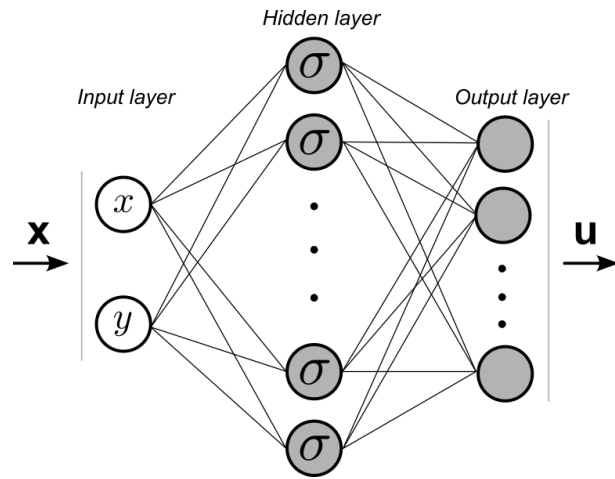
Figure 2: Neural Network: architecture, layers and neurons.
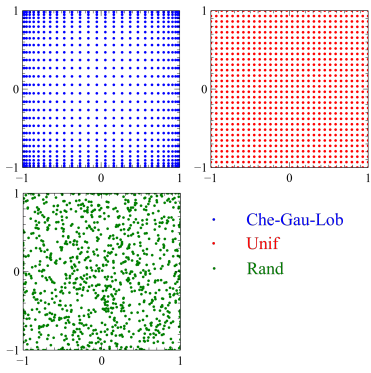
(a) Collocation points distribution.
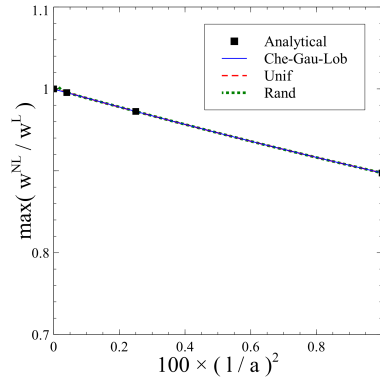
(b) Boundary conditions: SSSS.
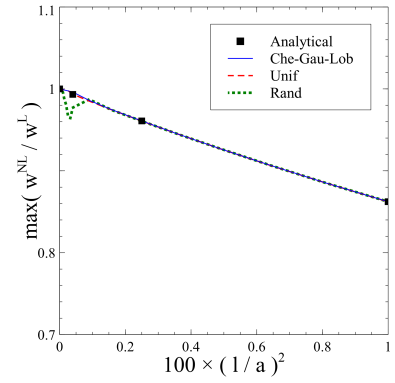
(c) Boundary conditions: SCSC.
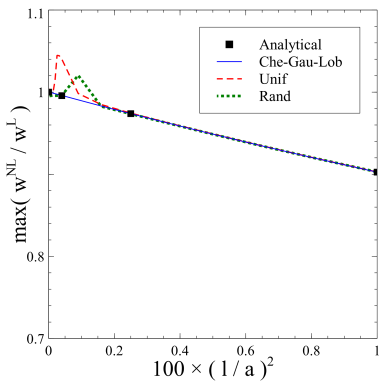
(d) Boundary conditions: SFSF.
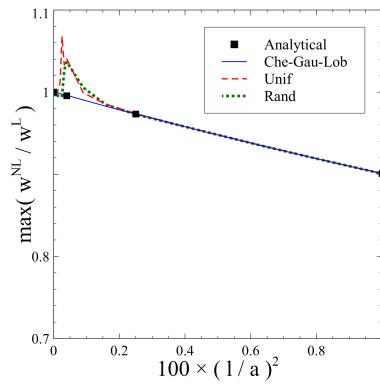
(e) Boundary conditions: SCSF.
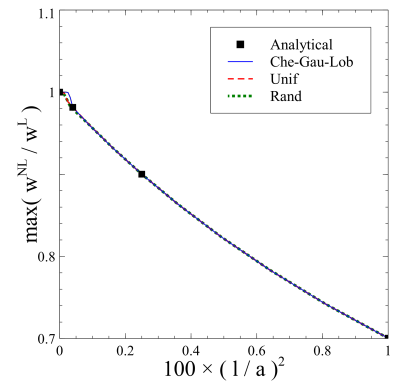
(f) Boundary conditions: CCCC.

Figure 3: Parametric study on distribution of collocation points.
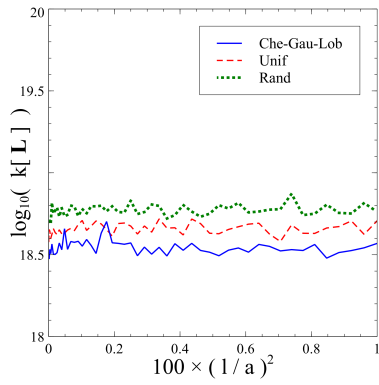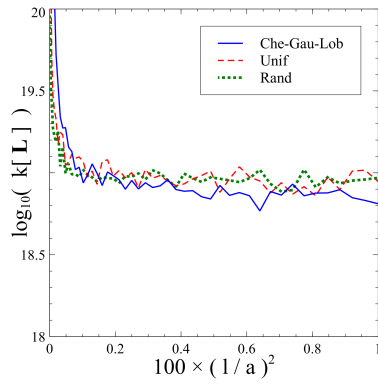
(a) Boundary conditions: SSSS.   (b) Boundary conditions: SCSC.   (c) Boundary conditions: CCCC.
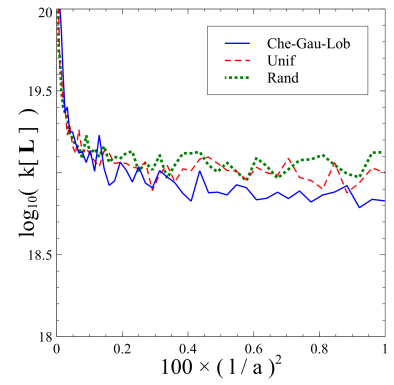
Figure 4: Dependency of the condition number of coefficient matrix on local paramater.
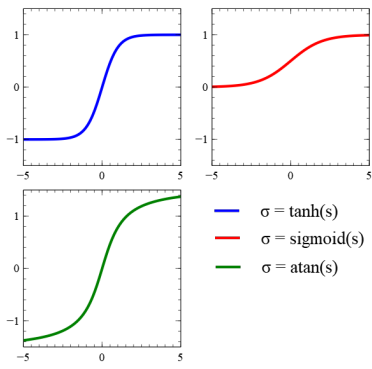
(a) Activation functions.
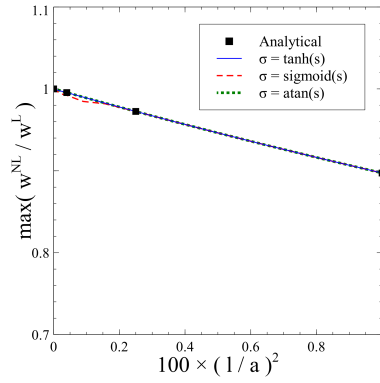
(b) Boundary conditions: SSSS.
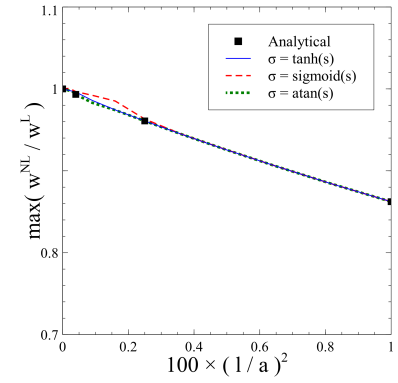
(c) Boundary conditions: SCSC.
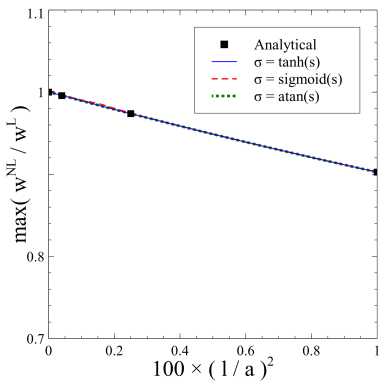
(d) Boundary conditions: SFSF.
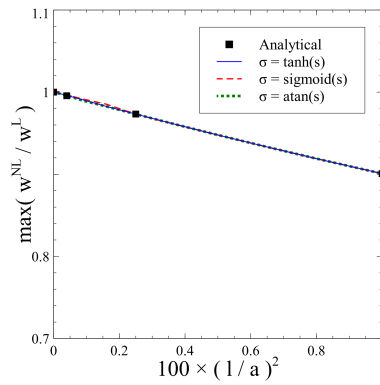
(e) Boundary conditions: SCSF.
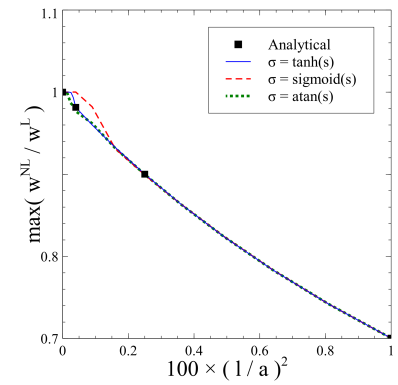
(f) Boundary conditions: CCCC.

Figure 5: Parametric study on activation functions.
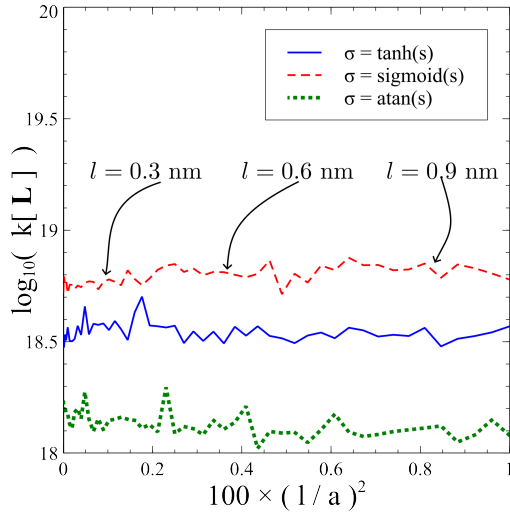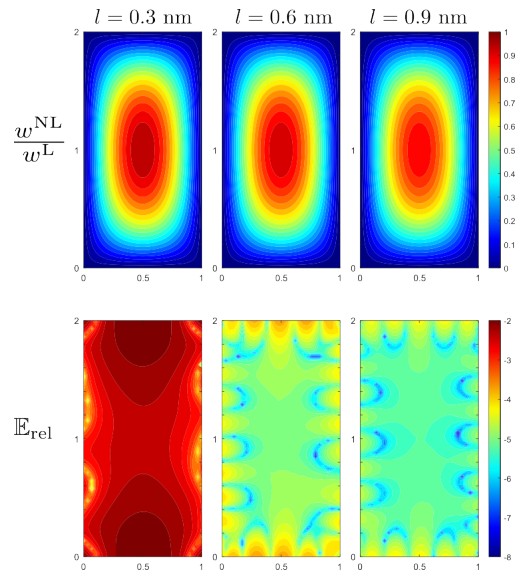
Figure 6: (a) Dependency of the condition number of coefficient matrix on local paramater for boundary conditions SSSS, and (b) solution outcome for different values of nonlocal parameter using the sigmoid activation function.
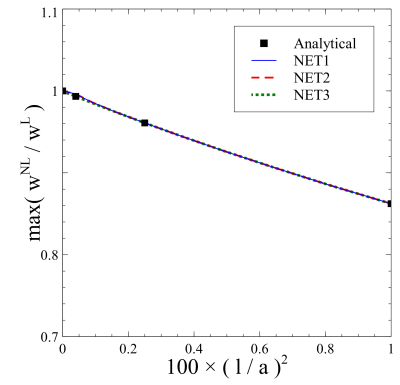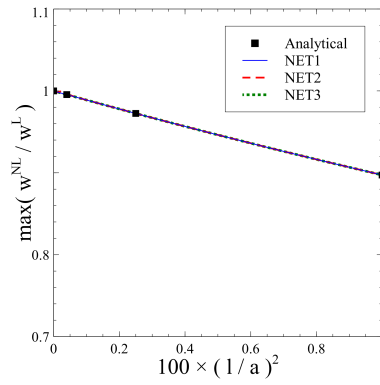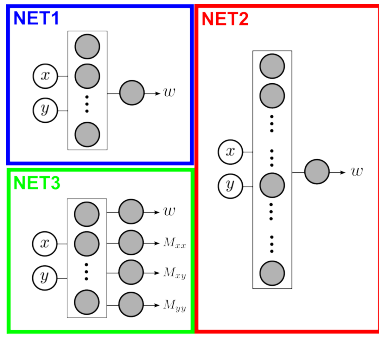
(a) Network architectures.

(b) Boundary conditions: SSSS.

(c) Boundary conditions: SCSC.
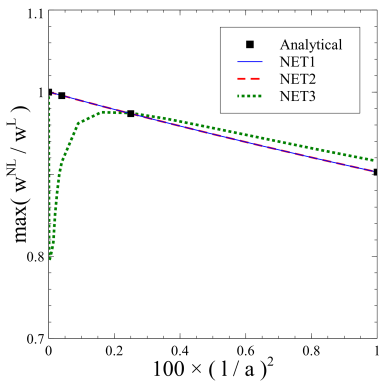
(d) Boundary conditions: SFSF.
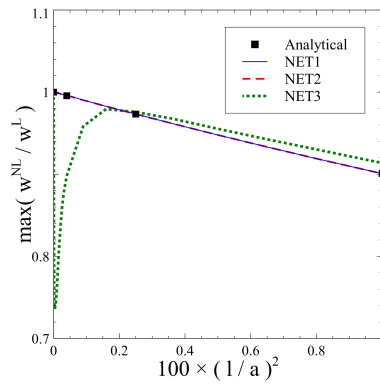
(e) Boundary conditions: SCSF.
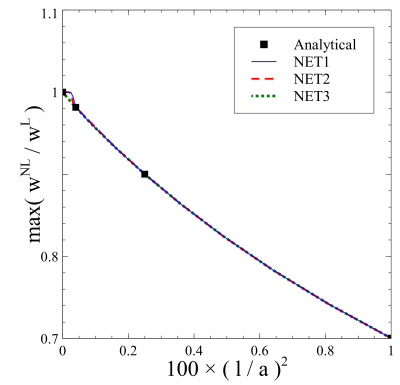
(f) Boundary conditions: CCCC.

Figure 7: Parametric study on network architectures.
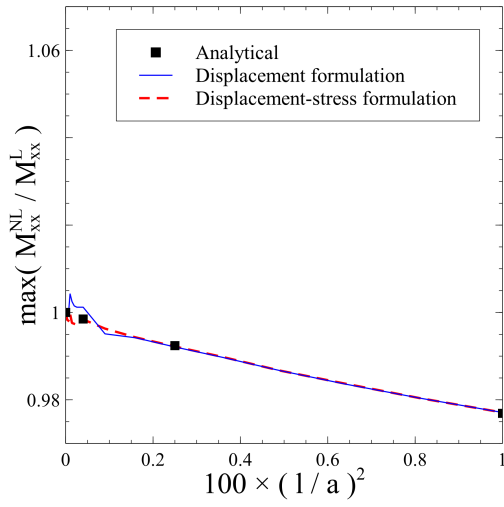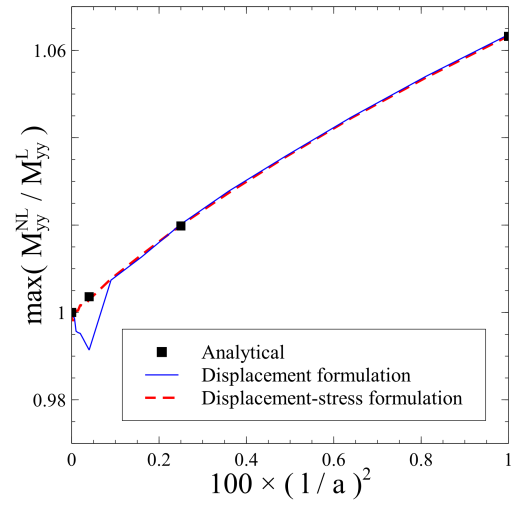
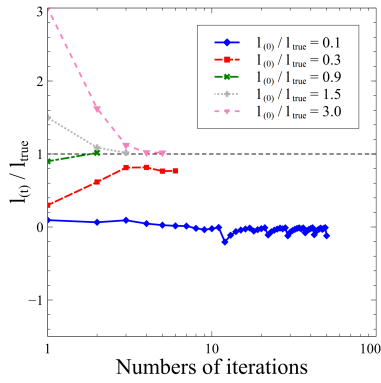Figure 8: Bending moments predictions for different values of nonlocal parameter: (a) M_{xx} and (b) M_{yy}.
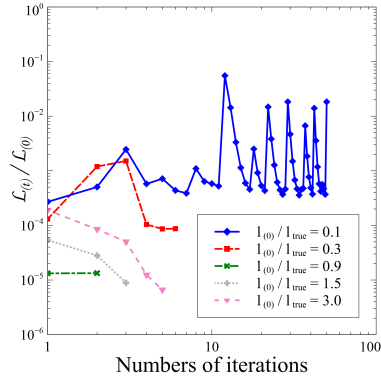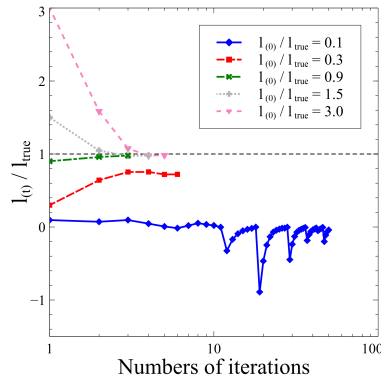
(a) Noise level: $n_\% = 0\%$.
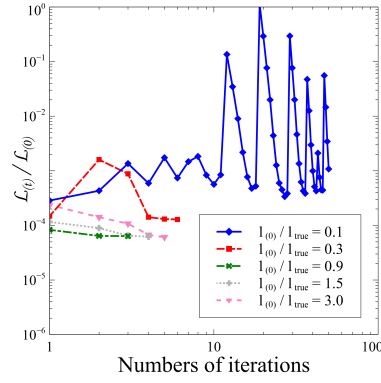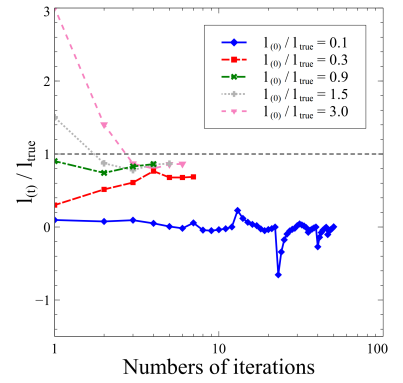
(b) Noise level: $n_\% = 1\%$.
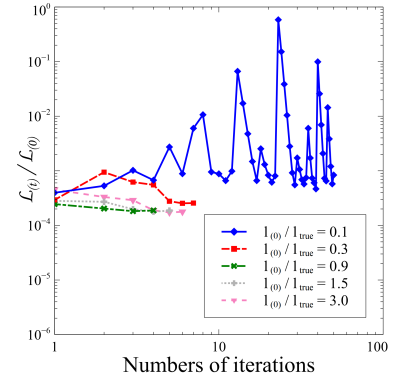
(c) Noise level: $n_\% = 3\%$.

(d) Noise level: $n_\% = 0\%$.

(e) Noise level: $n_\% = 1\%$.

(f) Noise level: $n_\% = 3\%$.

Figure 9: Evolution of the training process in terms of loss function ratio (a)-(c) and error fraction (d)-(f) for different levels of noise .