Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XL, págs. 1183-1194 (artículo completo)
F.A. Avid, L.C. Bessone, P. Gamazo, J.J. Penco, M.A. Pucheta, M.A. Storti (Eds.)
Concordia, 6-9 Noviembre 2023

EXPLORING THE CAPABILITY OF PINNS FOR SOLVING MATERIAL IDENTIFICATION PROBLEMS

C. Díaz-Cuadro, M. C. Vanzulli and P. Galione

Instituto de Ingeniería Mecánica y Producción Industrial, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Keywords: PINNs, Material Identification.

Abstract. Physics-Informed Neural Networks (PINNs) have emerged as a promising approach for solving scientific and engineering problems that involve partial differential equations or physical constraints. PINNs are a type of neural network architecture that incorporates physical laws or governing equations into its learning process. By combining the strengths of deep learning and physics-based modeling, PINNs can learn complex patterns and relationships from data while simultaneously satisfying the governing equations or physical laws. In this work, we explore the capabilities of PINNs to solve physical problems and identify the material properties. The first validation example is 1D problem, in which the heat generation number is estimated in a rectangular fin with temperature dependant thermal conductivity and heat generation. The second example illustrates the behavior of a 2D linear-elastic beam subjected to a uniform traction at its tip, experiencing negligible strains and plane stresses. The goal in this example was to estimate the Young's modulus. Finally, the third example studying here is a three-dimensional solid with a Neo-Hookean material, loaded with a compressive traction at the opposite end. In this case, the estimated parameters were the first and second Lame's parameters. The reliability of the results was assessed comparing against the analytical solution of each case. The ground truth displacement data were obtained from analytical solution of the problem evaluated in selected data points. These values were used as input to evaluate the loss data function, while the remaining loss functions were derived from the physics of each problem. The results of this study suggest that PINNs have the potential to be an effective tool for both material identification problems and real-time prediction of the physical solution.

1 INTRODUCTION

In the last decade, Deep Neural Networks (DNNs) have revolutionized various fields by exhibiting exceptional capabilities in pattern recognition, image processing, natural language understanding, and more (Krizhevsky et al., 2017; Alipanahi et al., 2015; Lake et al., 2015; Mallampati and Almekkawy, 2021). These networks, inspired by the architecture of the human brain, have showcased unparalleled prowess in capturing complex relationships within data, allowing them to excel in tasks that are considered challenging for traditional algorithms if enough data is available (Goodfellow et al., 2016). In this context, Physics-Informed Neural Networks (PINNs), which where first introduced in Raissi et al. (2019), emerged as a novel framework based on automatic differentiation where neural networks are not merely data-driven, but also informed by the governing physics equations. Automatic differentiation (Paszke et al., 2017) is a fundamental feature that allows the library to compute gradients or derivatives of functions with respect to their input variables automatically. By incorporating physical laws as constraints during the training process, PINNs aim to strike a balance between empirical data and fundamental principles, leading to enhanced generalization and accurate predictions, even with limited training data. This method offers several advantages over traditional solvers. One key benefit is its mesh-free nature, relying on collocation points that can be placed anywhere. Next, this architecture makes feasible the addition of physical knowledge (constraint, boundary condition, physical laws, observational data) about any part of a complex system into the network, this helps to guide the optimization to the solution and avoid unrealistic results (Fuhg and Bouklas, 2022).

The paper follows a structured format, with Section 2 describing the methodology. Subsection 2.1 introduces Deep Neural Networks (DNNs) while Subsection 2.2 explores Physics-Informed Neural Networks (PINNs). Subsections 3.1 to 3.3 provide theoretical foundations in thermodynamics of a fin, linear elasticity, and hyperelasticity. Section 3 presents numerical results, including in Subsections 2.3 - stationary fin, 2.4 - linear elastic cantilever beam, and 2.5 - hyperelastic uniaxial compression. The paper concludes in Section 4, summarizing findings and emphasizing the broader implications of this research.

2 METHDOLOGY

In this section, we elucidate the step-by-step procedures undertaken to achieve the study's objectives, encompassing experimental design, data collection and analysis methods, theoretical frameworks, and computational tools.

2.1 Deep feedforward neural networks

DNNs consist of interconnected units called neurons organized into layers. Fig. 1 illustrates a deep feedforward neural network composed of interconnected layers of neurons, which compute an output layer (predictions) based on input data. The information is propagated forward through the layers, creating a learning network with some form of feedback mechanism.

DNNs define a mapping between the output $\mathbf{y}$ and input $\mathbf{x}$ through the function $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$, and learns the value of the parameters $\boldsymbol{\theta}$ that result in the best function approximation. To find this $\boldsymbol{\theta}_{NN}$ for a parametric family of distributions $f(\mathbf{x}; \boldsymbol{\theta}_{NN})$ the negative log-likelihood estimation is minimized through an optimization process (Goodfellow et al., 2016).

The layers are connected through weights $\mathbf{W}$ and biases $\mathbf{b}$. For a particular layer l, the output

$\hat{\mathbf{Y}}^l$ can be expressed as is presented in Eq. :

$$\begin{aligned} \mathbf{Z}^l &= \mathbf{W}^l \hat{\mathbf{Y}}^{l-1} + \mathbf{b}^l, \\ \hat{\mathbf{Y}}^l &= \sigma(\mathbf{Z}^l) \end{aligned} \tag{1}$$

where σ represents the activation function applied to the intermediate output $\mathbf{Z}^l$.

2.2 Physics Informed Neural Network

The architecture selected for all the PINNs has a number of layers $N_L = 8$ with $N_{NL} = 20$ neurons each, as it is shown in Fig. 1. The first part of the PINN (a DNN) receives the reference position $\mathbf{X}$ as input and returns the predicted $\mathbf{u}_{NN}$. Now, consider the set of non-linear PDE equations presented in Eq. (2):

$$\Phi(\mathbf{x}, \mathbf{u}, \mathcal{D}\mathbf{u}, \mathcal{D}^2\mathbf{u}, \mathbf{c}) = \mathbf{0} \tag{2}$$

where $\mathcal{D}\mathbf{u}$, $\mathcal{D}^2\mathbf{u}$ represents gradient and Hessian of u respectively whereas Φ is generic nonlinear function. To solve Eq. (2), the second part of the PINN, takes $\mathbf{u}_{NN}$ as input and computes three types of losses, which are chosen as the Mean Squared Error between the corresponding values, as it is shown in Eqs. (3), (4), and (5). These losses are used backward to update the weights, biases and the trainable material properties $\mathbf{c}$. The required derivatives are computed using automatic differentiation. Three losses are considered. The data loss $\mathcal{L}_{data}$, which is evaluated at the data points point N_d with known θ values. The partial differential equation loss $\mathcal{L}_{PDE}$ (physics loss), is evaluated at N_{PDE} collocation points. The boundary condition loss $\mathcal{L}_{BC}$, evaluates at the boundary of the body, which is a point, a line, or a surface depending of the case. Each loss is computed as it is described in the following Eqs. (3), (4) and (5):

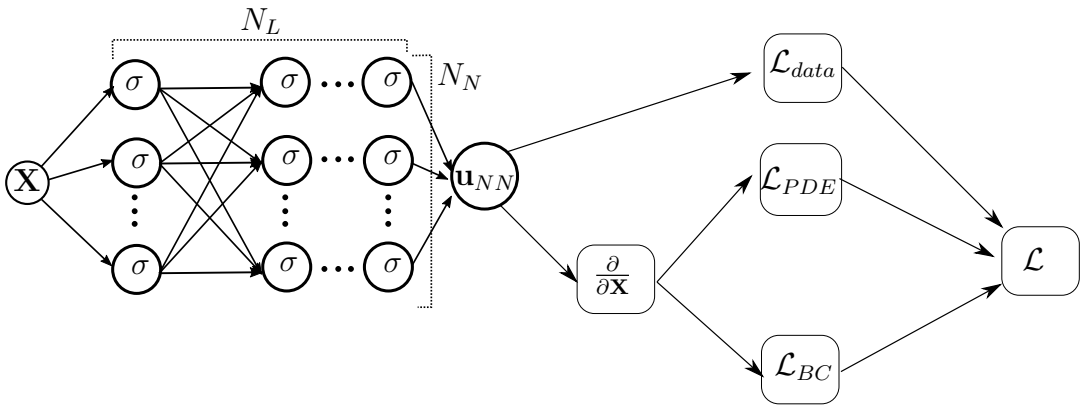


Figure 1: Sketch of the PINN implemented.

$$\mathcal{L}_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} ||\mathbf{u}^i_{data} - \mathbf{u}^i_{NN}||_2^2, \tag{3}$$

$$\mathcal{L}_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} ||\Phi(\mathbf{u}^i_{NN}, \partial_{\mathbf{x}}\mathbf{u}^i_{NN}, \partial_{\mathbf{xx}}\mathbf{u}^i_{NN})|_{\mathbf{X}^i_{PDE}}||_2^2, \tag{4}$$

$$\mathcal{L}_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} ||\Omega_{data}(\mathbf{X}^i_{BC}) - \Omega(\mathbf{u}^i_{NN}, \partial_{\mathbf{x}}\mathbf{u}^i_{NN}, \partial_{\mathbf{xx}}\mathbf{u}^i_{NN})|_{\boldsymbol{X}^i_{BC}}||^2_2. \tag{5}$$

$\Phi(\cdot)$ is the partial differential equations derived from the physics of each problem depending on the neural network output ($\mathbf{u}$), and its derivatives ($\partial_{\mathbf{x}}\mathbf{u}, \partial_{\mathbf{xx}}\mathbf{u}$). $\Omega(\cdot)$ is the boundary condition function and $\Omega_{data}(\cdot)$ is the ground truth. Each one of these functions is evaluated in the corresponding points, N_{PDE} and N_{BC} respectively.

Then, the total loss is computed using Eq. (6):

$$\mathcal{L} = \alpha_{data}\mathcal{L}_{data} + \alpha_{PDE}\mathcal{L}_{PDE} + \alpha_{BC}\mathcal{L}_{BC} \tag{6}$$

where the α_i is a weight associated to the loss i, which were manually tuned.

Loss weights are used to adjust the relative importance of these components in the optimization process. The choice of loss weights influences how much emphasis the model places on satisfying the data and physics constraints. Finding appropriate loss weights is crucial for achieving a well-balanced model that respects both data and physics Xiang et al. (2022).

For all cases, the optimization algorithm employed was L-BFGS (Liu and Nocedal, 1989), which is a second order algorithm. From this point, the tolerances set to solve with L-BFGS are 1×10^{-11} for the loss absolute change and its gradient.

2.3 Thermodynamic of a fin

This problem is based on Oommen and Srinivasan (2022), in which a rectangular fin with temperature dependant thermal conductivity and heat generation is considered. The governing equation of this problem is well known as it is presented in Eq. (7):

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) - \frac{hP}{A}(T - T_\infty) + q = 0 \tag{7}$$

where k is the thermal conductivity and q is the internal heat source, which in this case varies linearly with temperature. The non-dimensional form of Eq. (7) is presented as follows in Eq. (8):

$$F = \frac{\partial}{\partial x*}\left((1 + \epsilon_c\theta)\frac{\partial \theta}{\partial x*}\right) - N^2\theta + N^2 G(1 + \epsilon_G\theta) = 0 \tag{8}$$

with the following dimensionless numbers:

$$\theta = \frac{T - T_\infty}{T_b - T_\infty} \qquad N = \sqrt{\frac{hPL^2}{k_0 A}} \quad G = \frac{q_0 A}{hP(T_b - T_\infty)}$$
$$\epsilon_G = \epsilon(T_b - T_\infty) \qquad \epsilon_C = \beta(T_b - T_\infty) \tag{9}$$

where $\mathbf{c} = [G]$ is the heat generation number to be estimated, ϵ_C is the non-dimensional thermal conductivity parameter, ϵ_G is the non-dimensional heat generation parameter and N is the convection-conduction parameter.

2.4 Linear Elasticity

An elastic plate submitted to small strains and undergoing plane stresses can be modeled with the Hooke's Law. Hooke's Law is a foundational principle in linear elasticity, stating that

within the elastic limit of a material, stress is directly proportional to strain. The most common representation of this relationship is using the Hooke's law for linear elasticity, which can be expressed in the tonsorial form as it is presented in Eq. (10):

$$\boldsymbol{\sigma} = \mathcal{C} \cdot \boldsymbol{\varepsilon} \tag{10}$$

where: $\boldsymbol{\sigma}$ represents the stress tensor describing the distribution of stresses in the material, $\mathcal{C}$ is the fourth-order stiffness tensor known as the elastic stiffness tensor or elasticity tensor. It characterizes the material's response to stress and relates it to strain, and $\boldsymbol{\varepsilon}$ represents the strain tensor describing material deformation.

Plane stress is a simplification used in the analysis of materials and structures where the stress variation in one dimension is negligible compared to that of the other two dimensions. In this case, this assumption allow us to consider the stresses only in x and y direction.

Lamé's parameters (λ and μ), Young's modulus (E), and Poisson's ratio (ν) are interconnected material properties that describe the mechanical behavior of isotropic linear elastic materials. These properties are used to model how materials respond to external forces and deformations, and are related as follows in Eq. (11):

$$\lambda = \frac{E}{3(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)}. \tag{11}$$

The equilibrium equations for a 2D plate relate the internal stresses to the external loads applied to it. These equations ensure that the sum of forces and moments in each direction is zero, maintaining equilibrium. In the case of linear elasticity this can be mathematically expressed through Eq. (12):

$$\Phi = \nabla . \boldsymbol{\sigma} - \mathbf{F}_{ext} = 0 \tag{12}$$

where $\mathbf{F}_{ext}$ is the vector of external forces.

2.5 Hyperelasticity

A body made of a isotropic and homogeneous hyperelastic material under finite deformation is considered. The body occupies the solid region Ω_s. The mapping function φ of material points $\mathbf{X}$ from the reference configuration to the current configuration $\mathbf{x}$ is expressed in Eq. (13):

$$\mathbf{x} = \varphi(\mathbf{X}) = \mathbf{X} + \mathbf{u}. \tag{13}$$

The deformation gradient $\mathbf{F}$ is defined in Eq. (14), by computing the material derivatives of $\varphi(\mathbf{X})$.

$$\mathbf{F} = \nabla \varphi(\mathbf{X}) \tag{14}$$

where ∇ represents the material gradient operation.

The equilibrium equations are given by Eq. (15):

$$\begin{aligned} \nabla \cdot \mathbf{P} + \mathbf{f} = 0 \quad & X \in \Omega_s \\ \mathbf{u} = \hat{\mathbf{u}} \quad & X \in \Gamma_D \\ \mathbf{P}.\mathbf{n} = \hat{\mathbf{t}} \quad & X \in \Gamma_N \end{aligned} \tag{15}$$

where $\mathbf{P}$ represents the first Piola-Kirchhoff stress, $\mathbf{f}$ is the body force, $\nabla \cdot$ the divergence operator, and $\mathbf{n}$ the outward normal unit vector in the reference configuration. A set of external

traction forces $\hat{\mathbf{t}}$ represents the Neumann boundary conditions applied at Γ_N region, while $\hat{\mathbf{u}}$ accounts for the Dirichlet boundary conditions at region Γ_D.

The constitutive law for Hyper-elastic materials satisfies Eq. (16):

$$\mathbf{P} = \frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}} \tag{16}$$

where Ψ is the strain energy density function specific for each material model assumed.

3 NUMERICAL RESULTS

Within this section, the results of three examples with incremental complexity are presented. The first one is a 1D heat transfer problem, while the second and third are 2D and 3D solid problems, respectively. Furthermore, in the first two, one material parameter is identified, whereas in the 3D problem, two are identified. For the resolution of this problems we use 3.10 Python version and the machine learning framework PyTorch 1.13 (Paszke et al., 2019).

3.1 Example 1D: Stationary fin

The profile of temperatures in nine equispaced points of a rectangular fin with temperature dependant thermal conductivity and heat generation is considered for the evaluation of the $\mathcal{L}_{data}$. Eq. (8) depends on several variables: the non-dimensional thermal conductivity parameter is $\varepsilon_C = 0.4$, the non-dimensional heat generation parameter is $\varepsilon_G = 0.6$ and the convection-conduction parameter is $N = 1$.

Moreover, the boundary conditions considered for the $\mathcal{L}_{BC}$ are given by Eq. (17):

$$\theta(x^*)|_{x^*=0} = 0 \qquad \frac{\partial \theta(x^*)}{\partial x^*}|_{x^*=1} = 0 \tag{17}$$

where the first one represents the imposed (or known) temperature at $x^* = 0$ of the fin and the second one is the adiabatic end. These boundary condition were imposed in different ways. The Dirichlet boundary condition is set as input such that $\theta(x^*)|_{x^*=0}$ is zero, ensuring that the network effectively enforces the dimensionless temperature to be zero at $x^* = 0$ (only one point). The Neumann boundary condition is imposed as a loss that has to be minimized, therefore the network tend to modify the output to fulfill this condition (at one point). For the computation of the $\mathcal{L}_{PDE}$ the Eq. (8) is evaluated in 50 collocation points, and the weights of the losses α_i are all considered as 1.

The results are shown in the following figures. One the one hand, Fig. 2 shows that the predicted temperature distribution (blue dashed line) are in agreement with the data provided as input (golden dots). On the other hand, a continuous decrease of the total loss with the advance of the iterations is observed. Fig. 3 shows the evolution of the predicted heat generation number with the iterations. It can be seen that the G value predicted converges to the G value ground truth. Especifically, the value predicted by the PINN was $G_{NN} = 0.705$ while the real value was $G_{real} = 0.703$. This difference represents a 0.3% of error in the prediction.

3.2 Example 2D: Linear-Elastic Cantilever Beam

An homogeneous two-dimensional body that occupies a prismatic domain denoted by $\Omega = [0,\ L_x] \times [0,\ L_y]$, as depicted in Fig. 4 is studied. The dimensions of this domain are $L_x = 30$ mm, $L_y = 10$ mm.

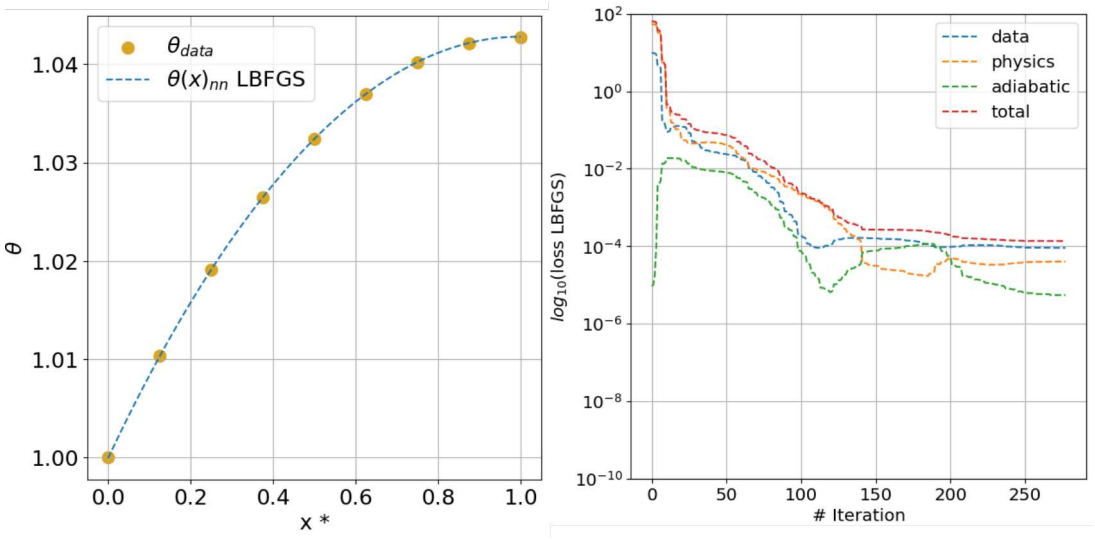


Figure 2: **left:** Temperature profile known vs predicted - **right:** Evolution of the losses for each L-BFGS iteration

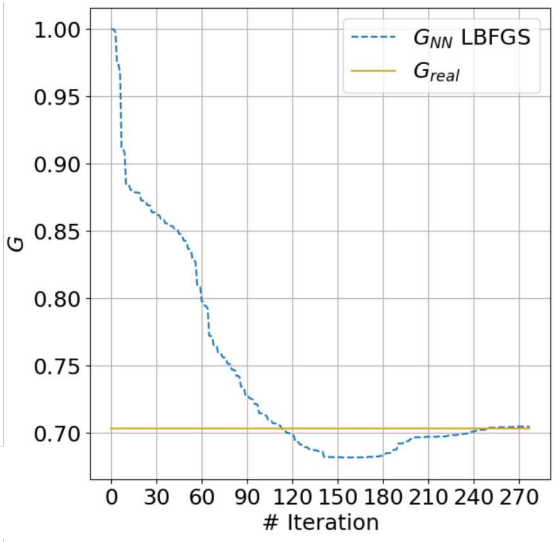


Figure 3: Evolution of the predicted heat generation with L-BFGS iterations

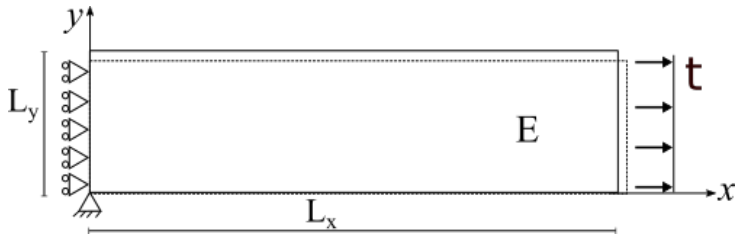


Figure 4: Plane stress traction sketch.

On the surface $\partial\Omega_{L_x}$ at $x = L_x$, a nominal total force, $\mathbf{t} = [1,\ 0,\ 0]$ N is applied uniformly over the surface. The boundary faces at $x = 0$ mm ($\partial\Omega_X$), $y = 0$ mm ($\partial\Omega_Y$) are considered to have frictionless contact. The remaining boundary ($\partial\Omega_{L_Y}$) is free and no tension is applied.

The mathematical expressions of the Dirchlet boundary conditions are given by Eq. (18):

$$\begin{aligned} \mathbf{u}(\mathbf{x}) \cdot (\mathbf{e}_1) = 0 \qquad \forall\, \mathbf{x} \in \partial\Omega_X \\ \mathbf{u}(\mathbf{x}) \cdot (\mathbf{e}_2) = 0 \qquad \forall\, \mathbf{x} \in \partial\Omega_Y \end{aligned} \tag{18}$$

and Neumann boundary conditions are presented in Eq.(19):

$$\mathbf{P}(\mathbf{x})[\mathbf{e}_1] = \mathbf{t} \qquad \forall\, \mathbf{x} \in \partial\Omega_{L_x} \tag{19}$$

$$\mathbf{P}(\mathbf{x})[\mathbf{e}_2] = \mathbf{0} \qquad \forall\, \mathbf{x} \in \partial\mathbf{\Omega}_{\mathbf{L_Y}} \tag{20}$$

As the body is considered submitted to a plane stress state, the constitutive law presented in Eq. (10) is simplified to Eq. (21) as follows:

$$\boldsymbol{\sigma} = G \begin{pmatrix} \dfrac{2}{1-\nu}(u_x + \nu v_y) & u_x + v_y \\ u_x + v_y & \dfrac{2}{1-\nu}(v_y + \nu u_x) \end{pmatrix}, \tag{21}$$

and substituting Eq. 21 into Eq. 12, the expression to use in the $\mathcal{L}_{PDE}$:

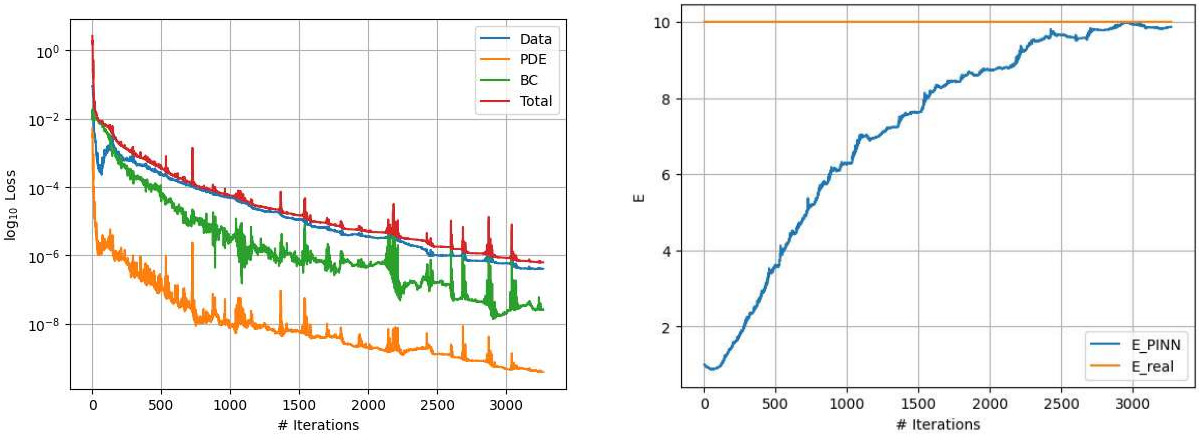
$$\Phi = \nabla.\boldsymbol{\sigma} - \mathbf{F}_{ext} = G \begin{pmatrix} \dfrac{2}{1-\nu}(u_{xx} + \nu v_{yx}) + u_{xy} + v_{yy} \\ \dfrac{2}{1-\nu}(v_{yy} + \nu u_{xy}) + u_{xx} + v_{yx} \end{pmatrix} - \mathbf{F}_{ext} = 0 \tag{22}$$

where G is the shear modulus which can be computed from E and ν as: $G = E/2(1+\nu)$. For this example, $\mathbf{C} = [E]$ is the parameter to be estimated by the PINN (which should be 10), while $\nu = 0.3$ is a known parameter.

In the reference configuration, 1000 training points are uniformed placed in the square domain. These points are used to evaluate the loss terms for the PDEs. Additionally, 300 training points are used to evaluate the loss for the displacement field. Finally, for the application of Dirichlet and Neumann boundary conditions (see Eqs.(18) and (19)) the strategy was the same than that in the previous example. For the Dirichlet boundary condition the zero displacement point were provided as input data, while for Neumann boundary condition, 30 points were defined at the sides of the 2D plate where the loss of the boundary condition ($\mathcal{L}_{BC}$) was evaluated. After several computations it was found that $\mathcal{L}_{PDE}$ is a few orders smaller than the $\mathcal{L}_{data}$, or even the $\mathcal{L}_{BC}$. In this case, the weights of the losses play a key role to make comparable the magnitudes. The weights α_i were manually tuned to $\alpha_{data} = 1$, $\alpha_{PDE} = 500$, and $\alpha_{BC} = 1$, and the learning rate was $3e^0$.

The results are shown in Fig. 5. During the training process, we concurrently test the performance of the PINN by evaluating both the losses and the predicted material parameters. The evolution of the total loss against the training iterations is shown in Fig. 5a. The total loss gradually decreases from $\mathcal{O}(10^0)$ to approx $\mathcal{O}(10^{-6})$ despite some fluctuations, indicating that the PDEs, BCs and the displacement data are all satisfied.

As the training proceeds, the inferred material parameters almost reaches the reference values ($E_{NN} = 9.86$ vs $E_{real} = 10$), representing in this case, the predicting error in the identification is around a 1.3%



(a) Evolution of losses functions.

(b) Evolution of material parameters E.

Figure 5: Plane stress example training process.

3.3 Example 3D: Hyperelastic Uniaxial Compression

An homogeneous body that occupies a prismatic domain denoted by $\Omega = [0,\ L_x] \times [0,\ L_y] \times [0,\ L_z]$, as depicted in Fig. 6 is solved. The dimensions of this domain are $L_x = 3$ mm, $L_y = 1$ mm, and $L_z = 1$ mm. On the surface $\partial\Omega_{L_i}$ at $x = L_x$, a nominal force per unit area, $\mathbf{t} = [-5,\ 0,\ 0]$ MPa is applied. The boundary faces at $x = 0$ mm ($\partial\Omega_X$), $y = 0$ mm ($\partial\Omega_Y$), and $z = 0$ mm ($\partial\Omega_Z$) are considered to have frictionless contact. The remaining two boundaries are free and no tension is applied. The mathematical expressions of the Dirchlet boundary conditions are given by Eq. (23):

$$\begin{aligned} \mathbf{u}(\mathbf{x}) \cdot (\mathbf{e}_1) = 0 \qquad & \forall\, \mathbf{x} \in \partial\Omega_X \\ \mathbf{u}(\mathbf{x}) \cdot (\mathbf{e}_2) = 0 \qquad & \forall\, \mathbf{x} \in \partial\Omega_Y \\ \mathbf{u}(\mathbf{x}) \cdot (\mathbf{e}_3) = 0 \qquad & \forall\, \mathbf{x} \in \partial\Omega_Z, \end{aligned} \tag{23}$$

and the Neumann boundary satisfies the following Eq. (24):

$$\mathbf{P}(\mathbf{x})[\mathbf{e}_1] = \mathbf{t} \qquad \forall\, \mathbf{x} \in \partial\Omega_{L_i} \tag{24}$$

$$\mathbf{P}(\mathbf{x})[\mathbf{n}_i] = \mathbf{0} \qquad \forall\, \mathbf{x} \in \partial\mathbf{\Omega_i} \tag{25}$$

where $\partial\Omega_i$ represents all the solid surfaces excluding $\partial\Omega_i$, $\partial\Omega_X$, $\partial\Omega_Y$, $\partial\Omega_Z$.

The constitutive material behaviour is a Neo-hookean with a strain energy function presented in Eq. (26):

$$\Psi(\mathbf{C}) = \frac{\mu}{2}(\mathrm{Tr}(\mathbf{C}) - 2 - \ln(J)) + \frac{K}{2}(J-1)^2 \tag{26}$$

where $\mathbf{C}$ is the Cauchy-Green strain tensor, $J = \sqrt{(\det(\mathbf{C})}$, μ is the second Lamé's parameter and K is the bulk modulus. In this example $\mathbf{c} = [K, \mu]$ and the reference values are $K = 1.538$ MPa and $\mu = 3.33$ MPa respectively.

The analytic solution of the problem can be obtained solving the following Eqs. (27) and (28) :

$$\mu - \frac{\mu}{\alpha^2} + \frac{K\beta^2}{\alpha}(\beta^2\alpha - 1) \;=\; -p(t) \tag{27}$$

$$\mu - \frac{\mu}{\beta^2} + K(\alpha^2\beta^2 - \alpha) \;=\; 0 \tag{28}$$

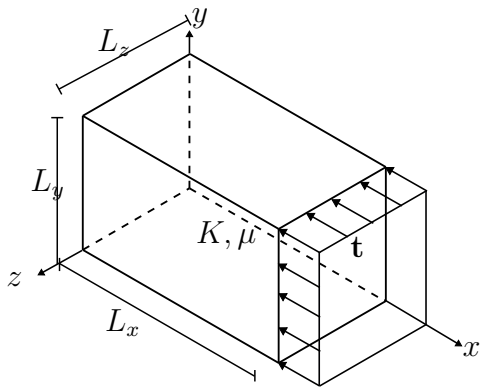


Figure 6: Uniaxial-compression sketch.

where $\alpha = (1+u_x/L_x)$, $\beta = (1+u_y/L_y)$ and u_x, u_y are the displacements at $\mathbf{x} = (L_x,\ L_y,\ L_z)$.

In the reference configuration, a total of 100 training points are placed randomly in the square domain. These points are used to evaluate the loss terms for the PDEs. Additionally, 25 training points are used to evaluate the loss for the displacement field. Finally, for Dirichlet and Neumann boundary conditions in Eqs.(23) and (24), 50 points randomly distributed at each face are employed to evaluate the training loss.

The results are shown in Figs. 7 and 8. During the training process, we concurrently test the performance of the PINN by evaluating the predicted material parameters in Fig. 8. As the training proceeds, the inferred material parameters reaches the reference values. In this case, the final identification of K and μ were 3.290 and 1.549 respectively, while the real values were 3.333 for K and 1.538 for μ. The values results in an identification error of 1.3% in K and 0.7% error in μ.

The evolution of the loses against the number of training iterations is shown in Fig. 7. The loses gradually decrease from $\mathcal{O}(10^1)$ to approx $\mathcal{O}(10^{-6})$ despite some fluctuations, indicating that the PDEs, BCs and the displacement data are all satisfied.

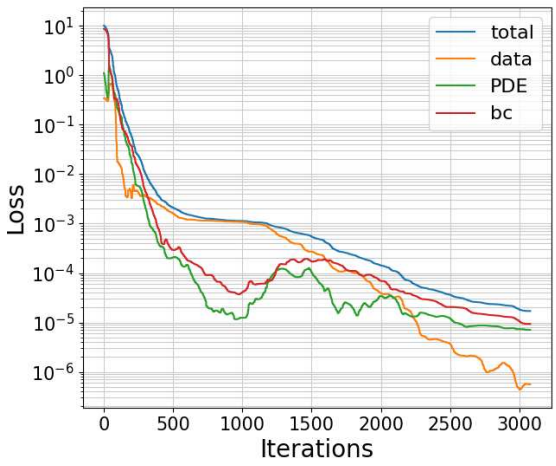


Figure 7: Evolution of losses functions.

The results obtained suggest that the PINN has sufficiently utilized the conditions from physics and data and accurately recovered the reference material parameters.

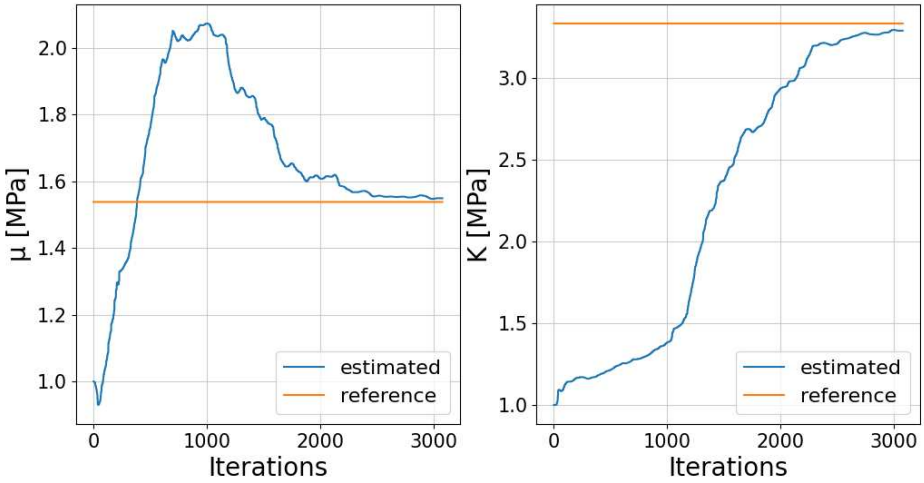


Figure 8: Evolution of material parameters K and μ.

4 CONCLUSIONS

In the three presented examples, the PINN was able to identify the material parameters with an error of less than 1.3% in all cases in only a few minutes, making it an interesting framework to be extended to more real and complex problems. One of the main findings was that once the PINN was implemented for the first studied case, extending it to the other analyzed cases was straightforward, requiring only the implementation of the corresponding equations to the physics of each particular problem. This allows for a considerably rapid extension to many problems where the solution to the inverse problem could be of great interest. Based on these results, it is concluded that PINNs are a versatile tool, easy to extend to a wide variety of cases with different numbers of input and output variables. Moreover, an interesting capability of PINNs is that they allow to incorporate as much constraints, conditions, and information as known at any point of the body, which can be a difficult (or even impossible) task in FEM.

Some disadvantages are that tuning loss weights manually has limitations. It's time-consuming, lacks systematic guidance, and is sensitive to initial choices. This can lead to suboptimal convergence and performance. More structured optimization strategies are needed to get better results. In that direction, gradient-based optimization techniques or utilizing hyperparameter optimization algorithms, could offer a more robust and efficient approach to fine-tune loss weights.

Moreover, despite that at first sight PINNs are a framework that does not require meshing the geometry, when Neumann boundary conditions are present it is necessary to inform the network about the normal vector to the surface at the points where this boundary condition is evaluated, which implies the need to provide the network with more information about the geometry.

There are several paths to explore as a future works, some of them being: material identification in more realistic cases in both material behavior and geometries; nonlinear behavior on materials with nonlinear mechanical responses, like elastoplastic or viscoelastic materials; application of PINNs to time-dependent systems, such as dynamic material properties or transient behaviors.

5 AKNOWLEDGMENTS

The authors would like to thank *Agencia Nacional de Investigación* for their financial support. The authors also want to thank Professors Martin Draper, Jorge Perez Zerpa from *Universidad de la República* and Vivek Omen, Kamraj Shukla from CRUNCH group of Brown University.

REFERENCES

Alipanahi B., Delong A., Weirauch M.T., and Frey B.J. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

Fuhg J.N. and Bouklas N. The mixed deep energy method for resolving concentration features in finite strain hyperelasticity. *Journal of Computational Physics*, 451:110839, 2022.

Goodfellow I., Bengio Y., and Courville A. *Deep learning*. MIT press, 2016.

Krizhevsky A., Sutskever I., and Hinton G.E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Lake B.M., Salakhutdinov R., and Tenenbaum J.B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Liu D.C. and Nocedal J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Mallampati A. and Almekkawy M. Measuring tissue elastic properties using physics based neural networks. In *2021 IEEE UFFC Latin America Ultrasonics Symposium (LAUS)*, pages 1–4. IEEE, 2021.

Oommen V. and Srinivasan B. Solving inverse heat transfer problems without surrogate models: A fast, data-sparse, physics informed neural network approach. *Journal of Computing and Information Science in Engineering*, 22(4):041012, 2022.

Paszke A., Gross S., Chintala S., Chanan G., Yang E., DeVito Z., Lin Z., Desmaison A., Antiga L., and Lerer A. Automatic differentiation in pytorch. 2017.

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Raissi M., Perdikaris P., and 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.

Xiang Z., Peng W., Liu X., and Yao W. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022.