Proc. of the 8th Intl. Symp. on Reliability Engineering and Risk Management (ISRERM 2022)
Edited by Michael Beer, Enrico Zio, Kok-Kwang Phoon, and Bilal M. Ayyub
©2022 ISRERM Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-5184-1_GS-07-026-cd

Identification of Material Parameters from Full-Field Displacement Data Using Physics-Informed Neural Networks

D. Anton *[1] and H. Wessels[1]

[1]*Institute for Computational Modeling in Civil Engineering, Technische Universität Braunschweig, Germany*

Abstract

For the condition assessment of existing infrastructure buildings the current parameters of the building material, such as the Young's modulus and Poisson's ratio, are of great interest. These parameters indicate damage or material degradation, since they reflect the resistance of the structures to external impacts. Provided the displacement field data is available, the material parameters can be identified by solving the momentum equation inversely. It was recently shown that there is a method in the growing field of scientific machine learning, known as physics-informed neural networks [1], that is particularly suitable for the inverse solution of partial differential equations. In contrast to purely data-driven approaches, physics-informed neural networks do not only use the displacement data but also the physics behind the data formulated as partial differential equation. It has been shown that physics-informed neural networks can in principle be used to identify material parameters [2], [3]. In the present paper, we first develop a three-stage adaptation of the method to realistic one-dimensional applications. Second, we verify the adapted method on two-dimensional displacement data. We conclude the paper with an outlook on ongoing and future work to further improve the method.

Keywords: Inverse problems, Parameter identification, Physics-informed neural networks, Deep learning

1. Introduction

Continuous structural health monitoring is the prerequisite of a reliable prediction of the remaining service life of infrastructure buildings [4], [5]. When designing new infrastructure buildings, it is assumed that both the building structure and material do not age during service life. This assumption is incorrect as in reality, the material ages due to chemical and physical processes. These processes in turn result in a decreasing resistance of the building structure to external impacts. At the same time, the external impact to the building structure, e.g. traffic load, often increases more than it was assumed during the design phase. By using monitoring systems, on the other hand, the actual condition of the infrastructure building can be determined. From the monitored evolution of the building structure condition and the impacts, a forecast of the remaining service life can then be made. In addition, the monitored evolution of the infrastructure condition can be used as a basis for decisions on maintenance or reinforcement measures.

Therefore, the identification of material parameters is of great interest for the condition assessment of existing infrastructure buildings. Material degradation usually leads to a decrease in the material stiffness that in turn results in a decreased resistance of the building structure. In the present paper, our investigations focus on steel structures and we assume linear-elastic material behaviour. Thus, the material condition of steel is, among others, directly reflected by Young's modulus and Poisson's ratio.

Provided that the displacement data can be measured, e.g. by digital image correlation (DIC), the material parameters of interest can be identified by solving an inverse problem. The underlying equation of this inverse problem is the momentum equation that is a fundamental physical principle. Since the momentum equation is a partial differential equation (PDE), identifying the material parameters from the momentum equation can be formulated as a PDE constrained optimization problem.

While numerical methods such as the finite element method (FEM) have proven successful for the forward solution of PDEs, these methods often reach their limits for inverse problems. In [6], two different approaches for solving inverse problems using FEM were compared: First, the parameter field to be identified was bound to the discretization of the domain used in FEM. Second, the parameter field was globally represented by an artificial neural network (ANN). Even thought, the authors conducted only a qualitative comparison, some capabilities of ANNs and some limitations of FEM for solving inverse problems were demonstrated. It could be shown, that ANNs are insensitive to noise and could better deal with incomplete data.

In the meanwhile, it was recently shown that physics-informed neural networks (PINNs) [1] are particularly suitable for solving PDEs inversely. The idea behind this method goes back to the 1990s [7], [8]. However, it has become applicable only recently due to developments in automatic differentiation [9], advanced software frameworks and libraries for machine learning applications and more powerful hardware such as GPUs and TPUs. In contrast to purely data-driven approaches, instead of using only measurement data, PINNs also make use of prior knowledge. This prior knowledge includes the physical laws behind our observations which are known to us in many engineering problems. In addition, PINNs can also be applied to noisy measurement data as well as to high dimensional problems.

In engineering and science, PINNs have already been applied to inverse problems from versatile fields, including the simulation of unsaturated groundwater flow [10], biomechanics [11], nano-optics and meta materials [12] as well as damage mechanics [13], to name only a few examples. Also in the field of solid mechanics, PINNs were used to identify model parameters for linear-elastic and nonlinear-elastoplastic material models from synthetic data [2]. In contrast to the present work, the authors have assumed that for solving the inverse problem, apart from the displacement data, also stress data is available. Additionally, in the linear-elastic case the Lamé constants were set to $\lambda = 1.0$ and $\mu = 0.5$. The assumptions made for the

*E-Mail: anton@irmb.tu-bs.de

Lamé constants are not realistic, when the material at hand is, e.g., steel with a Young's modulus of $210,000\ \frac{N}{mm^2}$ and a Poisson's ratio of 0.3. Furthermore, in [3], PINNs were already applied to identify the parameter of an inhomogeneous, incompressible, hyperelastic material only from displacement data. Since the investigated material was inhomogeneous, for the shear modulus a parameter field has to be learned which was approximated by another ANN. However, even in this application, the true shear modulus was in the range $[0.15, 0.37]$ which is not comparable to the realistic shear modulus of steel. Both for the investigations in [2] and [3], the domain considered was a unit square. In summary, the previous works have shown that PINNs can in principle be successfully applied to inverse problems and in particular for material parameter identification. However, the simplifying assumptions made in the literature are not comparative to the real-world inverse problem, where the material parameters of steel are to be identified from real DIC data.

Previous investigations of our group have focused on PINNs for solving forward problems [14], [15]. The purpose of this work is to adapt the method to the identification of the material parameters of steel from DIC data for the linear-elastic case. It is demonstrated that the method fails without further adaptions in real-world problems in structural mechanics. The reason for this is that the inverse problem needs to be conditioned on at least three different stages. For these three stages, we develop conditioning approaches for realistic one-dimensional applications and verify the effectiveness of the adapted method using synthetic displacement data.

The remainder of this paper is organized as follows: In sec. 2 the momentum equation and the governing equations of linear elasticity are reviewed. Sec. 3 provides a brief introduction to physics-informed neural networks (PINNs) for solving inverse problems. In sec. 4, first the typical failure modes that occur when applying PINNs to realistic displacement data are identified. Based on our observations, the method is then step by step adapted and verified using realistic one-dimensional displacement data. In sec. 5 we apply the adapted method to two-dimensional displacement data. Finally, we conclude our investigations in sec. 6 and point out directions of further research.

2. Momentum equation and governing equations of linear elasticity

Material parameters can be identified from displacement data by solving the momentum equation inversely. The strong form of the momentum equation can be derived as

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x} \in \Omega. \tag{1}$$

Here, ∇ denotes the nabla operator, $\boldsymbol{\sigma}(\boldsymbol{x})$ the stress tensor and $\boldsymbol{f}(\boldsymbol{x})$ a vector field describing the volume forces. The solution of eq. 1 must satisfy the boundary conditions and the strong form of the PDE at all points $\boldsymbol{x}$ in the domain Ω.

The constitutive equation for linear-elastic, homogeneous material is defined as

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \lambda\, tr(\boldsymbol{\varepsilon}(\boldsymbol{x}))\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{2}$$

where $tr()$ is the trace operator and $\boldsymbol{I}$ the second order identity tensor. λ and μ are Lamé's first and second constants depending on the Young's modulus E and the Poisson's ratio ν by

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

The strain tensor $\boldsymbol{\varepsilon}(\boldsymbol{x})$ can be calculated from the displacement vector field $\boldsymbol{u}(\boldsymbol{x})$ as follows

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \frac{1}{2}(\nabla \boldsymbol{u}(\boldsymbol{x}) + \nabla \boldsymbol{u}(\boldsymbol{x})^T). \tag{4}$$

Hence, the momentum equation provides the relation between the displacement field, i.e., its second partial derivatives, and the material parameters which are Young's modulus and Poisson's ratio.

3. Physics-informed neural networks

3.1. Artificial neural networks are universal function approximators

Artificial neural networks (ANNs) are global, smooth function approximators. Provided that an ANN has a sufficient number of parameters, it can approximate any continuous function to an arbitrarily small error [16], [17]. By training the ANN, its parameters are adjusted with the aim that the ANN approximates the hidden function as well as possible. Thus, the training of ANNs is an optimization problem where a cost function has to be minimized. In supervised learning, this cost function gives a measure of how well the predictions of the ANN match the output of the labelled training data. For optimization, usually gradient based optimization algorithms are used. The gradient of the loss function according to the ANN parameters can be obtained using automatic differentiation [9].

An ANN consists of a high number of neurons which are the computational units, typically arranged in an input, an output and any number of hidden layers. In a fully connected feed-forward ANN, the neurons of each two successive layers are connected while each connection is given a weight. In addition, a bias is assigned to all neurons in the hidden layers and the output layer. Weights and biases compose the trainable parameters of a feed-forward ANN. The schematic structure of an ANN is shown in fig. 1. From a mathematical point of view, an ANN is a highly parameterized, composed function that defines a mapping $\mathbb{R}^N \rightarrow \mathbb{R}^M$ from an input vector to an output vector.

In all neurons except input neurons, the activity of the neuron is calculated from the weighted input to the neuron. While usually sigmoid functions, rectified linear units or the hyperbolic tangent are chosen as activation function in hidden neurons, in regression tasks, linear functions are often used in the output neurons.

For further explanations, we now introduce the notation applied in the following. We consider an ANN where layer 0 is the input layer and layer L the output layer. This ANN consists of $L+1$ layers in total and $L-1$ hidden layers. The bias of neuron j in layer l is denoted as b_j^l. Considering a whole layer, the biases of all neurons of layer l can be combined in a vector denoted as $\boldsymbol{b}^l$. Following

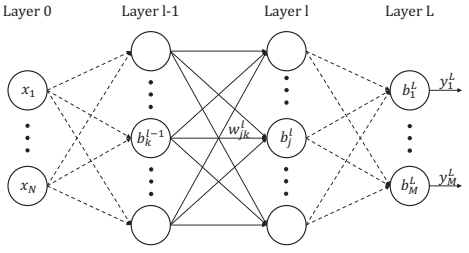


Figure 1. Schematic representation of a fully connected feed-forward ANN according to [6].

this principle, the weight of the connection between neuron k in layer $l-1$ and neuron j in layer l is denoted as w_{jk}^l, whereas all weights for the connections between layer $l-1$ and l can be combined in the weight matrix $\boldsymbol{W}^l$. Throughout this paper, within each layer the same activation function is used. The activation function in layer l is then denoted as σ^l for a single neuron and $\boldsymbol{\sigma}^l$ for the whole layer. Finally, the output of the neuron j in layer l is y_j^l and the output of layer l can be combined in $\boldsymbol{y}^l$.

According to this notation we can define the ANN output as a function of the input in a recursive formulation. The weighted input of neuron j in layer l is defined as

$$z_j^l = \sum_k w_{jk}^l y_k^{l-1} + b_j^l, \; l \neq 0, \tag{5}$$

where y_k^{l-1} in eq. 5 is given by

$$y_k^{l-1} = \sigma^{l-1}(z_k^{l-1}). \tag{6}$$

By inserting eq. 6 to eq. 5 and rewriting the result in symbolic notation, we obtain

$$\boldsymbol{z}^l = \boldsymbol{W}^l \boldsymbol{\sigma}^{l-1}(\boldsymbol{z}^{l-1}) + \boldsymbol{b}^l, \; l \neq 0, \tag{7}$$

where $\boldsymbol{\sigma}^{l-1}$ is applied elementwise.

Starting from the ANN output $\boldsymbol{y}^L$ the recursive definition terminates with the input vector $\boldsymbol{x}$. Since we do not use any activation function in the input layer, for the input layer applies

$$\boldsymbol{\sigma}^0(z^0) = \boldsymbol{x}. \tag{8}$$

Given eq. 5 - 8, the output of a feed-forward ANN as a function of the input $\boldsymbol{x}$ can ultimately be defined recursively as follows:

$$\begin{aligned}
\boldsymbol{y}^L &= \boldsymbol{\sigma}^L(\boldsymbol{z}^L) \\
\boldsymbol{z}^L &= \boldsymbol{W}^L \boldsymbol{\sigma}^{L-1}(\boldsymbol{z}^{L-1}) + \boldsymbol{b}^L \\
\boldsymbol{z}^{L-1} &= \boldsymbol{W}^{L-1} \boldsymbol{\sigma}^{L-2}(\boldsymbol{z}^{L-2}) + \boldsymbol{b}^{L-1} \\
&\vdots \\
\boldsymbol{z}^2 &= \boldsymbol{W}^2 \boldsymbol{\sigma}^1(\boldsymbol{z}^1) + \boldsymbol{b}^2 \\
\boldsymbol{z}^1 &= \boldsymbol{W}^1 \boldsymbol{x} + \boldsymbol{b}^1.
\end{aligned} \tag{9}$$

For a more in-depth introduction to deep learning, please refer to standard textbooks, e.g. [18].

3.2. Incorporating prior knowledge results in better generalization performance

In contrast to purely data-driven approaches, physics-informed neural networks (PINNs) are not trained by using only labelled training data but also by leveraging available prior knowledge [1]. In the forward problem, the physical laws are known and encoded in a PDE, but the solution of this PDE is unknown. The ANN, on the other hand, is a function approximator and acts as an approximation of the PDE solution. By adding regularizing terms to the loss function which further constrain the solution space, the ANN is enforced to satisfy the governing PDE. Ultimately, incorporating prior knowledge does not only reduce the amount of required data but also results in better generalization performance [19].

In the following, we consider a time-independent, nonlinear PDE parameterized by $\boldsymbol{\lambda}$ and defined on $\Omega \subset \mathbb{R}^d$ with suitable boundary conditions on $\partial\Omega$ with the general form

$$f(\boldsymbol{x}, \frac{\partial u}{\partial x_1}, ..., \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, ..., \frac{\partial^2 u}{\partial x_1 \partial x_d}; ...; \boldsymbol{\lambda}) = 0, \quad \boldsymbol{x} \in \Omega, \tag{10}$$

where $u(\boldsymbol{x})$ is the hidden solution of eq. 10. The boundary conditions could be Dirichlet, Neumann, Robin or periodic boundary conditions.

The inclusion of prior knowledge succeeds by embedding the governing PDE in the loss function. For solving the PDE, the first step is to approximate the hidden solution by an ANN $\hat{u}(\boldsymbol{x}, \boldsymbol{\theta})$. In the next step, the loss function of the forward problem in its general form is defined as

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &:= \mathcal{L}_r(\boldsymbol{\theta}) + \mathcal{L}_{d_b}(\boldsymbol{\theta}) + \mathcal{L}_{d_o}(\boldsymbol{\theta}) \\
\mathcal{L}_r(\boldsymbol{\theta}) &= \frac{1}{N_r} \sum_{k=1}^{N_r} [r]^2 \\
\mathcal{L}_{d_b}(\boldsymbol{\theta}) &= \frac{1}{N_b} \sum_{k=1}^{N_b} [\hat{u}(\boldsymbol{x}_b^k; \boldsymbol{\theta}) - u_b^k]^2 \\
\mathcal{L}_{d_o}(\boldsymbol{\theta}) &= \frac{1}{N_o} \sum_{k=1}^{N_o} [\hat{u}(\boldsymbol{x}_o^k; \boldsymbol{\theta}) - u_o^k]^2
\end{aligned} \tag{11}$$

with the PDE residual calculated as

$$r = f(\boldsymbol{x}, \frac{\partial \hat{u}}{\partial x_1}, ..., \frac{\partial \hat{u}}{\partial x_d}; \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_1}, ..., \frac{\partial^2 \hat{u}}{\partial x_1 \partial x_d}; ...; \boldsymbol{\lambda}). \tag{12}$$

Here, $\{x_b^k, u_b^k\}_{k=1}^{N_b}$ and $\{x_o^k, u_o^k\}_{k=1}^{N_o}$ are the sets of labelled training data for the boundary conditions and observations, respectively. When balancing of the loss function becomes necessary, the loss terms in eq. 11 can additionally be weighted by weights λ_r, λ_{d_b} and λ_{d_o}, respectively. To calculate the residual r from eq. 12, the partial derivatives of the ANN $\hat{u}(\boldsymbol{x}, \boldsymbol{\theta})$ can be determined using automatic differentiation. The forward problem is then solved by searching the parameters $\boldsymbol{\theta}^*$ of the ANN that minimize the loss function defined in eq. 11. Finally, approximating

the hidden PDE solution by an ANN following the PINN approach described above is an optimization problem which can be formulated as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{13}$$

In inverse problems, the objective is usually the identification of the PDE parameters $\boldsymbol{\lambda}$. Either the solution of the PDE is known or at least some observations are available in the form of a set of data samples $\{x_o^k, u_o^k\}_{k=1}^{N_o}$. Therefore, for solving inverse problems, two changes must be made to the previous formulation. First, the loss function defined in eq. 11 is usually reduced to include only $\mathcal{L}_r(\boldsymbol{\theta})$ and $\mathcal{L}_{d_o}(\boldsymbol{\theta})$, while the latter term contains all observational data. This observational data also includes boundary data and we no longer distinguish between the data sets $\{x_b^k, u_b^k\}_{k=1}^{N_b}$ and $\{x_o^k, u_o^k\}_{k=1}^{N_o}$. Second, the PDE parameters become trainable parameters [1].

4. Method adaption to realistic one-dimensional applications

Previous work, which include [2], [3] among others, have proven the feasibility of material parameter identification from displacement data using PINNs. However, the assumptions made for this purpose often cannot be transferred to real-world applications, e.g. material parameter identification of steel from digital image correlation (DIC) data. These assumptions concern the domain size as well as the magnitude of displacements and material parameters. It is shown, that without further adaptions of the method, PINNs show only poor performance on parameter identification for realistic conditions. In the further course of this section we develop a three-stage adaption of the method and demonstrate its effectiveness by using synthetic but realistic one-dimensional displacement data.

For demonstration purposes, we assume to have displacement data of an one-dimensional stretched steel rod. The considered stretched rod has a length of $100\,mm$ and a cross sectional area of $100\,mm^2$. While the upper end of the stretched rod is clamped, the free end is pulled with a force of $1\,kN$. In addition, a volume force of $0.1\,\frac{N}{mm^3}$ acts on the entire length of the stretched rod. A sketch of the geometry and boundary conditions is shown in fig. 2.

In the one-dimensional case, the momentum equation from eq. 1 can be simplified to

$$\frac{\partial \sigma(x)}{\partial x} + f(x) = 0, \tag{14}$$

where $\sigma(x)$ is the stress. After substituting the stress by the constitutive relation, we obtain

$$E\frac{\partial^2 u(x)}{\partial x^2} + f(x) = 0. \tag{15}$$

Here, E is the Young's modulus and $u(x)$ the displacement of the stretched rod.

For the forward solution, we used realistic material parameters and set the Young's modulus to $210,000\,\frac{N}{mm^2}$. Based on the analytical solution of the forward problem,

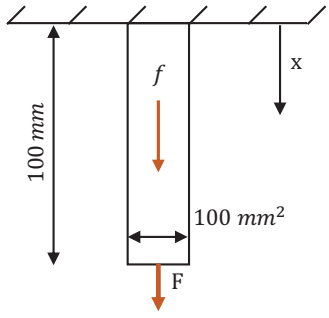


Figure 2. Geometry and boundary conditions of a stretched rod under tension load. The distributed load at the free end is set to $1\,kN$ and the volume force acting in the whole volume to $0.1\,\frac{N}{mm^3}$.

the displacement data for the inverse problem were generated. The displacement data set consists of 100 data samples uniformly sampled in the domain $[0\,mm, 100\,mm]$ with the coordinate as input and a related displacement in the range $[0\,mm, 0.007\,mm]$ as output. At the coordinates used in the training data set, the loss for the PDE residual is also determined during training.

Apart from the setup, the hyperparameters of both PINN and training algorithm have to be specified. To approximate the displacement of the stretched rod, we use a fully connected feed-forward ANN as defined in subsec. 3.1 with two hidden layers, each with 20 neurons. The hyperbolic tangent was used as activation function in the hidden layers and a linear function in the output layer. During the training, the ANN parameters and the unknown material parameters were optimized in full-batch mode using the Adam optimization algorithm [20] and a learning rate of 0.001. Training is limited to $20,000$ epochs, where one epoch is an iteration over the complete training data set. Before training starts, the weights were initialized using the *Glorot normal initialization* [21] and the biases are initialized with zeros. It is noted here that the ANN parameters are initialized identically for all simulations in this section. As long as it is not otherwise stated, the Young's modulus is initialized with zero. Since the hyperparameters have proven to be suitable for the following demonstration, no further fine tuning is conducted within this section.

Applying the PINN without further adaptions to the inverse problem described above yields a very poor approximation of the displacement field, c.f. fig. 3. Moreover, a Young's modulus of $17.15\,\frac{N}{mm^2}$ is predicted, which is far from the correct value. In the following, we develop a three-stage adaption of the method to the realistic one-dimensional application.

4.1. Normalization of inputs and outputs

Beside the poor result for the parameter identification, fig. 3 underlines that the Vanilla PINN is not capable to approximate the displacement from the provided data samples. As long as the approximation and its second derivative are not sufficiently accurate, the parameter cannot be identified correctly either. It is well known that convergence of

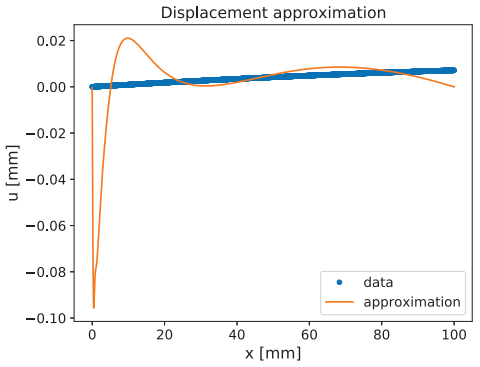


Figure 3. Vanilla PINN: Approximation of displacement data by PINN. Beside the poor approximation, the identified Young's modulus is far from the exact value.

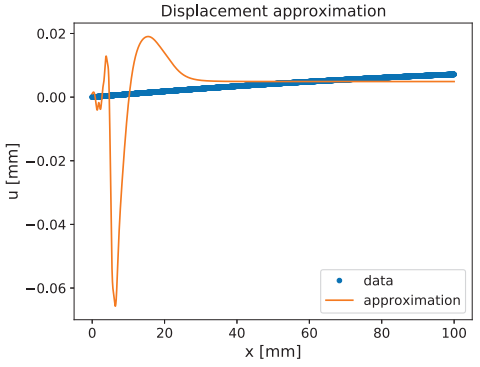


Figure 4. Normalized inputs and outputs: Approximation of displacement data by PINN. Normalizing the inputs and outputs is not sufficient to obtain a reasonable accurate approximation of the displacements, even if slight improvements can be noticed.

ANN training can be accelerated by normalizing the inputs. According to [22], the mean values of all input features should each be close to zero.

We apply this heuristic by normalizing both the input and output data from the original ranges to the range $[-1, 1]$. This is done by a linear transformation of the input and output spaces based on the minimum and maximum values, respectively. It is important to emphasize that the transformation of the inputs takes place after the real inputs are fed into the PINN. Likewise, the PINN outputs the real outputs after a retransformation of the predicted transformed outputs. As a result, the PINN does no longer learn the mapping from the real coordinates to the real displacements, but a mapping from the transformed inputs to the transformed outputs. As fig. 4 demonstrates, the effects of the normalization of both inputs and outputs alone are not sufficient. Also slight improvements are noticeable, further adaptions are still necessary.

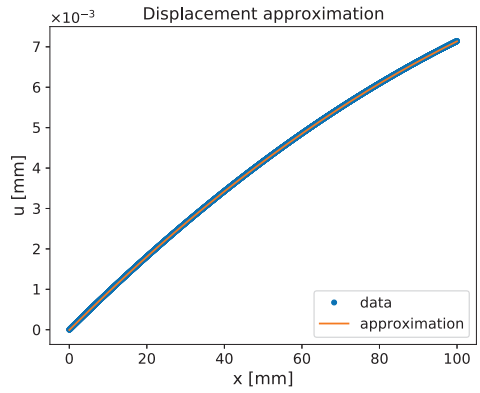


Figure 5. Conditioning of loss function: Approximation of displacement data by PINN. Conditioning the loss function results in a PINN approximation matching the measured data very well. Nevertheless, after normalizing the input and output as well as conditioning the loss function, the PINN is still not able to correctly identify the material parameter.

4.2. Conditioning the loss function

We identified another failure mode in an ill-conditioned loss function. The displacement data used in our simulations has a maximum displacement in the order of magnitude of $10^{-3}\,mm$. With displacement data in this range, even large relative deviations of the PINN from the measured displacement data result in only small losses in the loss term $\mathcal{L}_{d_o}(\boldsymbol{\theta})$. For illustration, a data sample with a measured displacement of $0.001\,mm$ is considered. If we now assume that the PINN deviates by 100% from the measured displacement, then according to eq. 11 this only results in a loss of 10^{-6} for the loss term $\mathcal{L}_{d_o}(\boldsymbol{\theta})$. The very small loss value, in turn, directly results in a very small gradient of the loss according to the parameters to be learned. At least in gradient based optimization algorithms, small gradients have limited effect on parameter optimization.

Instead of using mean squared error as metric for the loss term $\mathcal{L}_{d_o}(\boldsymbol{\theta})$, we introduce the relative mean squared error. The normalization of the deviation between the PINN and the training data is algorithmically realized by modifying the loss term $\mathcal{L}_{d_o}(\boldsymbol{\theta})$ from eq. 11 as follows

$$\mathcal{L}_{d_o}(\boldsymbol{\theta}) = \frac{1}{N_o}\sum_{k=1}^{N_o}\left[\frac{\hat{u}(\boldsymbol{x}_o^k;\boldsymbol{\theta}) - u_o^k}{l_{char}}\right]^2, \tag{16}$$

where l_{char} is the characteristic length. To ensure that even relatively small deviations again have an impact on the optimization, we propose to choose the maximum displacement from the training data set as the characteristic length. As fig. 5 shows, with the conditioned loss function, an accurate approximation of the measured displacement data can be obtained. Nevertheless, the PINN has identified a value of $18.4\,\frac{N}{mm^2}$ for Young's modulus and is thus still unable to correctly identify the material parameter.

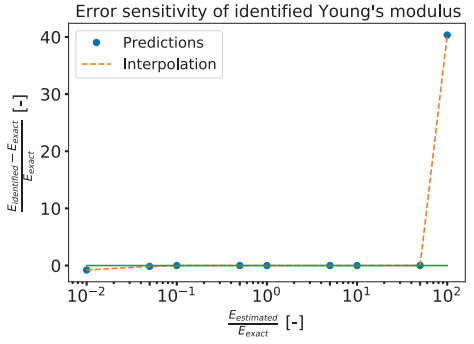


Figure 6. Error sensitivity of identified Young's modulus to initial estimate. For this purpose, Young's modulus was determined for different initial estimates in the range $[2.1 \cdot 10^3 \frac{N}{mm^2}, 2.1 \cdot 10^7 \frac{N}{mm^2}]$. The results show that the error is not sensitive to the estimate for realistic values for the Young's modulus.

4.3. Initial parameter estimation

The results obtained in the previous subsection show that although the PINN can approximate the displacement with high accuracy, it still cannot identify the material parameter. We found the reason for this in an ill-posed optimization problem. While the Young's modulus is initialized with $0.0 \frac{N}{mm^2}$, the parameter must have a value close to $210,000 \frac{N}{mm^2}$ at the end of a successful optimization. In comparison, the parameters of the ANN after training are in the range $[-0.73, 0.67]$ and have a mean value of -0.02. We therefore assume that the material parameter encounters many local minima on the long gradient path and gets stuck in one of them. In order to avoid this failure mode and to get a better posed optimization problem, we introduce an initial parameter estimate of Young's modulus. The underlying PDE from eq. 15 then changes to

$$\alpha_E E_{est} \frac{\partial^2 u(x)}{\partial x^2} + f(x) = 0. \tag{17}$$

Here, α_E is the correction factor initialized with 1.0 and E_{est} the initial estimate. With the introduction of an initial parameter estimate, the PINN does no longer learn the Young's modulus itself but a correction factor. By simply multiplying the correction factor with the initial estimate we get the identified Young's modulus. With the third adaption and an initial estimate of $210,000 \frac{N}{mm^2}$, we succeed in identifying a value of $E = 210,032.5 \frac{N}{mm^2}$ from the displacement data. The relative error from the exact value is less than 0.015%.

Finally, we analysed the sensitivity of the identified Young's modulus as a function of the initial parameter estimate. For this purpose, the Young's modulus was determined for different initial estimates in the range $[2.1 \cdot 10^3 \frac{N}{mm^2}, 2.1 \cdot 10^7 \frac{N}{mm^2}]$. The results in fig. 6 illustrate that the error is not sensitive to the estimate within a reasonable range.

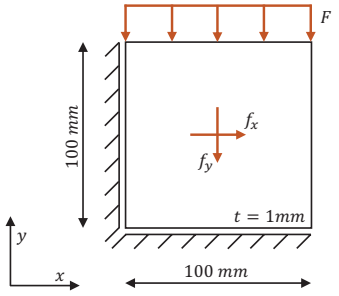


Figure 7. Geometry and boundary conditions of a rectangular plate in *plane stress* condition under load. The distributed load is set to 100 N and the volume forces acting in the whole volume to $1.0 \frac{N}{mm^3}$ and $-1.0 \frac{N}{mm^3}$ in dimensions x and y, respectively.

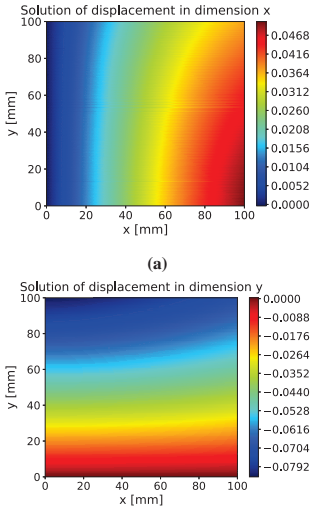
5. Application of adapted method to two-dimensional displacement data

We have shown in the previous section that after a three-staged adaption, PINNs can accurately identify Young's moduli from one-dimensional displacement data of a linear-elastic stretched rod. In the following, we verify that PINNs can also be used for material parameter identification from two-dimensional displacement fields.

For generation of the synthetic data set we considered a rectangular plate in *plane stress* condition. The geometry of the plate as well as the boundary conditions are outlined in fig. 7. The displacement field is induced by a combination of a distributed load of 100 N and volume forces with an absolute amount of $1.0 \frac{N}{mm^3}$ acting throughout the whole volume. Up to now, we need volume forces in both directions, otherwise the PINNs learn the trivial solution. Just as in the previous section, we choose the Young's modulus and the Poisson's ratio to be $210,000 \frac{N}{mm^2}$ and 0.3, respectively. Based on a FEM solution shown in fig. 8, we generated a synthetic data set consisting of a total of 65×65 data samples on an uniform grid. Each data sample is composed of the coordinates in dimensions x and y as inputs and the displacements $u_x(x, y)$ and $u_y(x, y)$ as outputs.

Some adjustments to the hyperparameters are required when applying PINNs to two-dimensional displacement fields. In order to solve the momentum equation in two dimensions, we choose two independent ANNs to approximate the displacement fields $u_x(x, y)$ and $u_y(x, y)$. According to [2], using separate ANNs for the two displacement fields results in a far more effective strategy than approximating both displacement fields with one sufficiently wide ANN with two output neurons. Due to the increasing complexity of the displacement fields to be approximated, we increase the number of trainable parameters and use ANNs with two layers of 40 neurons each. We apply the three adaptions developed in sec. 4 and choose the maximum displacement as characteristic length for conditioning of the loss function, which is 0.084 mm. As the initial estimate, we provide the exact material parameters. The other hyperparameters are chosen as specified in sec. 4.

The mean relative L^2-Norm of the approximated displacement fields $u_x(x, y)$ and $u_y(x, y)$ is $1.3 \cdot 10^{-3}$ at the



(a)

(b)

Figure 8. Solution of displacement fields a) $u_x(x, y)$ and b) $u_y(x, y)$.

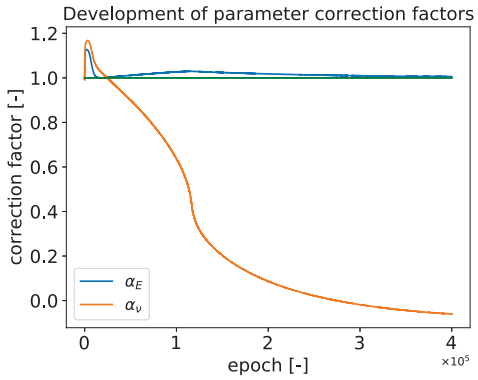


Figure 9. Development of predicted correction factors of Young's modulus and Poisson's ratio. The exact material parameters were used as the initial estimates. Therefore, if the material parameters were identified exactly, the correction factors would have to converge to 1.0.

end of the training. We could not observe any other anomalies in the approximated displacement fields. The development of the predicted correction factors for both Young's modulus and Poisson's ratio during training are shown in fig. 9. After training 400,000 epochs, the prediction for α_E and α_ν are 1.005 and −0.06, respectively. We observed similar results for volume forces with an absolute amount of 0.1 $\frac{N}{mm^3}$ and 10.0 $\frac{N}{mm^3}$.

We assume one possible reason for the large error in the two-dimensional setting in an ill-posed inverse problem. The ill-posedness of the inverse problem is caused by the Young's modulus which is in the order of magnitude of $O(10^5)$ $\frac{N}{mm^2}$. On the other hand, the volume forces in our example have a magnitude of 1.0 $\frac{N}{mm^3}$. As a consequence, the partial second derivatives of the displacement field are very small. At the same time, large relative errors in the second derivatives have a strong impact on the accuracy of the material parameters to be identified. This in turn results in high requirements for the accuracy of the displacement field approximation. To obtain a better posed problem, we are currently working on possibilities to further condition the momentum equation.

The simulation results underline that the parameter identification requires higher accuracy of the displacement field approximation. In the one-dimensional case, the relative L^2-Norm of the displacement approximation has a magnitude of $O(10^{-5})$ and the identification of the Young's modulus succeeds with a small relative error. In comparison, we find a large error in the identified Poisson's ratio in the two-dimensional case. At the same time, however, the relative L^2-Norm of the displacement field approximation has a magnitude of $O(10^{-3})$ and the accuracy is thus significantly lower than in the one-dimensional case. Therefore, our research additionally focuses on investigating methods for improving the accuracy of the approximated displacement field and its second partial derivatives, such as regularization techniques.

Another reason for the observed instabilities in optimization could be the used optimization algorithm. In order to be able to exclude this cause, we will also investigate other optimization algorithms in the future, including second-order optimization algorithms.

6. Conclusion

We developed a three-staged adaption of physics-informed neural networks (PINNs) for material parameter identification to realistic one-dimensional applications considering linear-elastic materials. First, we applied a normalization of both inputs and outputs of the artificial neural networks (ANNs). Second, after identification of another failure mode in too small gradients, we conditioned the loss function by normalizing the relatively small residuals in the data fit loss terms by a characteristic length which we set to the maximum displacement. Third, we enabled the PINN to identify the material parameters by providing an initial estimate. With this adaption, instead of the material parameter itself, the adapted PINNs learned a correction factor of the initial estimate.

Since we faced some problems when applying the adapted method to two-dimensional displacement data, we are currently working on further improving the method. Our ongoing and future research involves, among others, investigating what properties the displacement field must exhibit in order for material parameter identification from displacement data to be successful. In this context, we also consider more complex geometries, such as a plate with a hole. As we assume one reason for the large error in parameter identification in an ill-posed problem, we investigate possible solutions for obtaining a better posed inverse problem. Related to this, we found that the accuracy of the material parameter identification and the displacement field approximation are correlated. Therefore, our work

also focuses on how to increase the accuracy of both the displacement field approximation and its derivatives.

When it comes to real-world applications where the displacement data is measured by digital image correlation (DIC) another challenge is that the material parameters are stochastic in nature. Reasons are, among others, material inhomogeneities and noisy sensor data. Because of the significance for real applications, our ongoing research also aims to quantify the effect of the aforementioned uncertainties on the predicted material parameters. For this purpose, instead of the original PINNs, extended variants of them could be used, as proposed in [23] or [24].

Acknowledgment

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GRK2075/2 - Project number 255042459.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[2] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, 2021.

[3] E. Zhang, M. Yin, and G. E. Karniadakis, "Physics-Informed Neural Networks for Nonhomogeneous Material Identification in Elasticity Imaging," *arXiv Preprint*, p. 10, 2020. DOI: arXiv:2009.04525v1[cs.LG].

[4] A. Entezami, *Structural Health Monitoring by Time Series Analysis and Statistical Distance Measures*, ser. SpringerBriefs in Applied Sciences and Technologies. Cham: Springer, 2021, 145 pp., ISBN: 978-3-030-66259-2.

[5] F.-K. Chang, "Report on the First Stanford Workshop on Structural Health Monitoring," Stanford, AFRL-SR-BL-TR-98-0563, 1998.

[6] J. Berg and K. Nyström, "Neural networks as smooth priors for inverse problems for PDEs," *Journal of Computational Mathematics and Data Science*, vol. 1, 2021. DOI: 10.1016/j.jcmds.2021.100008.

[7] D. C. Psichogios and L. H. Ungar, "A Hybrid Neural Network-First Principles Approach to Process Modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.

[8] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[9] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic Differentiation in Machine Learning: A Survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 2018.

[10] I. Depina, S. Jain, S. M. Valsson, and H. Gotovac, "Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow," *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 2021. DOI: 10.1080/17499518.2021.1971251.

[11] W. Li and K.-M. Lee, "Physics informed neural network for parameter identification and boundary force estimation of compliant and biomechanical systems," *International Journal of Intelligent Robotics and Applications*, vol. 5, pp. 313–325, 2021. DOI: 10.1007/s41315-021-00196-x.

[12] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Optics Express*, vol. 28, no. 8, pp. 11 618–11 633, 2020. DOI: 10.1364/OE.384875.

[13] C. J. G. Rojas, M. L. Bitterncourt, and J. L. Boldrini, "Parameter Identification for a Damage Model Using a Physics Informed Neural Network," *arXiv Preprint*, p. 31, 2021. DOI: arXiv:2107.08781v1[cond-mat.mtrl-sci].

[14] H. Wessels, C. Weißenfels, and P. Wriggers, "The neural particle method – An updated Lagrangian physics informed neural network for computational fluid dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 368, 2020. DOI: 10.1016/j.cma.2020.113127.

[15] A. Henkes, H. Wessels, and R. Mahnken, "Physics informed neural networks for continuum micromechanics," *arXiv Preprint*, p. 28, 2021. DOI: arXiv:2110.07374v1[cs.LG].

[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[17] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, 798 pp. [Online]. Available: http://www.deeplearningbook.org (visited on 02/11/2020).

[19] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021. DOI: 10.1038/s42254-021-00314-5.

[20] D. P. Kingma and J. L. Ba, "ADAM : A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, 2015.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, Sardinia, 2010, pp. 249–256.

[22] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. B. Orr, K.-R. Müller, and G. Montavon, Eds., 2nd ed., Berlin: Springer, 2012, pp. 9–48, ISBN: 978-3-642-35288-1.

[23] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," *Journal of Computational Physics*, vol. 397, 2019. DOI: 10.1016/j.jcp.2019.07.048.

[24] L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data," *Journal of Computational Physics*, vol. 425, 2021. DOI: 10.1016/j.jcp.2020.109913.