Computers and Structures 283 (2023) 107054

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Coupling physics-informed neural networks and constitutive relation error concept to solve a parameter identification problem

Y. Wei [a], Q. Serra [a], G. Lubineau [b], E. Florentin [a,*]

[a] INSA Centre Val de Loire, Univ. Orléans, Univ. Tours, Laboratoire de Mécanique Gabriel LaMé, EA 7494, F-18022 Bourges, France
[b] King Abdullah University of Science and Technology, Mechanical Engineering Program, Physical Sciences and Engineering Division, Mechanics of Composites for Energy and Mobility Lab., Thuwal 23955-6900, Saudi Arabia

ARTICLE INFO

Article history:
Received 21 December 2022
Accepted 18 April 2023
Available online 3 May 2023

Keywords:
Identification
Physics-informed neural networks
Constitutive equation gap method
Inverse problem

ABSTRACT

Identification of material model parameters using full-field measurement is a common process both in industry and research. The constitutive equation gap method (CEGM) is a very powerful strategy for developing dedicated inverse methods, but suffers from the difficulty of building the admissible stress field. In this work, we present a new technique based on physics-informed neural networks (PINNs) to implement a CEGM optimization process. The main interest is to easily construct the admissible stress thanks to automatic differentiation (AD) associated with PINNs. This new method combines the high quality of the CEGM with the numerical effectivity of the PINNs and realizes the identification of material properties in a more concise way. We compare two variants of the developed method with the classical identification strategies on simple two-dimensional (2D) cases and illustrate its effectiveness in three-dimensional (3D) problems, which is of interest when dealing with tomographic images. The results indicate that the proposed method has good performance while avoiding complex calculation procedures, showing its great potential for practical applications.

1. Introduction

In the last decade, experimental mechanics for identifying the parameters of constitutive models have developed rapidly due to the popularity of full-field measurement techniques such as digital image correlation [1] or grid methods [2]. These techniques allow greater flexibility and provide very rich experimental data when applied to tests conducted under heterogeneous conditions [3]. Hence, effective inverse strategies are required to identify material parameters from the full-field measurements. Until now, many powerful strategies have been proposed, including the Finite Element Model Updating (FEMU) [4], the Equilibrium Gap Method (EGM) [5,6], the Virtual Fields Method (VFM) [7], the CEGM [8–10], etc. The advantage of CEGM over other methods is that it can be applied to any constitutive model and is not restricted by full-field measurements [11]. Although proven effective, CEGM suffers from the challenging calculation of the admissible stress field, especially when tackling 3D data. As a result, other techniques should be explored to avoid intricate implementation and to increase computational efficiency. Fortunately, deep learning methods provide us with new ideas for solving this problem.

Deep learning has achieved remarkable success in many fields [12], such as image processing and speech recognition. However, purely data-driven deep learning techniques are hardly used in practical engineering because the algorithms not only require large amounts of data but also contain a lot of randomnesses which will lead to a lack of confidence. In the last five years, integrating deep learning with already-known physics has become a popular solution, especially for adding physics knowledge to the training process of the algorithms [13]. Many approaches based on this idea have emerged to address different engineering and mathematical problems. E and Yu proposed the Deep Ritz Method for solving variational problems [14]. Sirignano and Spiliopoulos presented a mesh-free deep learning algorithm called the Deep Galerkin Method for solving the high-dimensional partial differential equations (PDEs) [15]. Guo et al. proposed the Deep Collocation Method for the bending analysis of Kirchhoff plate [16]. More recently, Nguyen-Thanh et al. developed the Deep Energy Method for nonlinear deformation hyperelasticity, which used potential energy as the loss function [17]. Raissi et al. have made a series of contributions in this field, and a typical example is PINNs [18]. The PINN algorithm can be used to approximate the solutions of PDEs by embedding them into the loss of neural network. By manipulating the loss function, this algorithm achieves an effect similar to regularization, thus avoids the local minimum problem to a certain

* Corresponding author.
E-mail address: eric.florentin@insa-cvl.fr (E. Florentin).

https://doi.org/10.1016/j.compstruc.2023.107054
0045-7949/

extent. Like other deep learning methods, PINNs can be a mesh-free method with the help of AD and it can overcome the curse of dimensionality [19–21]. Furthermore, in terms of implementation, PINNs can solve the inverse problem as easily as the forward problem. Recently, PINNs have been applied to various types of PDEs [22–24]. Since PDEs play a crucial role in engineering and fundamental sciences, PINNs are also used in many fields, for instance, fluid mechanics [25–27] and solid mechanics [28–31].

In particular, PINNs are also used to learn constitutive relationships or unknown parameters in PDE models [32,33], and some recent works apply this technology to identify material parameters [34–36]. In this paper, we present a new method by introducing the constitutive relation error concept into PINNs. Then we use it to identify the constitutive parameters of linear elastic material models. The proposed method constructs the loss function according to CEGM, applies the PINN algorithm to perform the originally intricate optimization process, and then identifies Young's modulus and Poisson's ratio of the material. Compared to CEGM, the developed method avoids complex calculations associated with a statically admissible stress field with the aid of AD, and the construction of admissible stress and the minimization are performed simultaneously. In contrast to other recent works, the introduction of the new concept makes the method possible to be extended to nonlinear behavior. In addition, we use novel neural network architectures for multi-parameter identification and compare the method with classic strategies. We also apply this method to the material identification of 3D objects. Regarding the implementation of algorithms, the proposed method can deal with 3D problems as easily as 2D problems, which is a significant advantage over traditional strategies.

The rest of paper is organized as follows. In Section 2, we explain the definition of the identification problem and its governing equations. In Section 3, after briefly introducing the PINNs, we propose two variants of the PINN-based identification methods. In Section 4, we present numerical experiments including comparisons with classical approaches. In Section 5, we demonstrate the capability of the proposed method in 3D problems. Finally, conclusions follow in Section 6.

2. Identification problem

Let us consider a generic elastic solid Ω of boundary $\partial\Omega$. Assuming the boundary conditions are defined by surfaces S_f and S_u, which satisfy $S_f \cup S_u = \partial\Omega$ and $S_f \cap S_u = \varnothing$. $\bar{\boldsymbol{T}}$ is the traction field prescribed over surface S_f, and $\bar{\boldsymbol{u}}$ is the displacement field prescribed over S_u. The equilibrium of Ω is governed by three sets of differential equations.

The first set of equations is the classical equilibrium equations, which are deduced by the equilibrium of forces:

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma} = 0 & \text{in } \Omega \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{T}} & \text{on } S_f \end{cases} \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress and $\boldsymbol{n}$ the outward unit normal to S_f, and div represents the divergence operator.

The second set is the kinematic compatibility equations that reflect the connection between displacement and strain:

$$\begin{cases} \boldsymbol{\varepsilon}[\boldsymbol{u}] = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla^t \boldsymbol{u}\right) & \text{in } \Omega \\ \boldsymbol{u} = \bar{\boldsymbol{u}} & \text{on } S_u \end{cases} \tag{2}$$

where $\boldsymbol{\varepsilon}[\boldsymbol{u}]$ is the strain tensor; ∇ and ∇^t denote the gradient and its transpose, respectively.

The third set is the constitutive equations, which reflect the stress–strain relationships via the generalized Hooke equation:

$$\boldsymbol{\sigma} = \mathscr{A} : \boldsymbol{\varepsilon}[\boldsymbol{u}] \quad \text{in } \Omega \tag{3}$$

where $\mathscr{A}$ is the constitutive parameters or Hooke's operator. In the context of heterogeneous isotropic elasticity, $\mathscr{A}$ is space-dependent and can be described as two independent parameters: Young's modulus E and Poisson's ratio ν, or Lamé constants λ and μ.

A common problem in solid mechanics is to calculate the displacement $\boldsymbol{u}$, stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$ given the geometry Ω, constitutive parameters $\mathscr{A}$ and boundary conditions $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{T}}$. This type of problem is referred to as direct or forward. For a direct problem, two classical definitions of spaces are usually introduced. Firstly, the kinematically admissible displacement fields are defined by:

$$\mathscr{C} = \{\boldsymbol{v} \in \mathscr{V}_u, \ \boldsymbol{v} = \bar{\boldsymbol{u}} \ (\text{over } S_u)\} \tag{4}$$

where $\mathscr{V}_u = \left[H^1(\Omega)\right]^3$ is the space where the displacement field is being sought. Secondly, the statically admissible stress fields are defined by:

$$\mathscr{S} = \left\{\boldsymbol{\tau} \in \mathscr{V}_\sigma, \ \operatorname{div} \boldsymbol{\tau} = 0 \ (\text{in } \Omega) \text{ and } \boldsymbol{\tau} \cdot \boldsymbol{n} = \bar{\boldsymbol{T}} \ (\text{over } S_f)\right\} \tag{5}$$

where $\mathscr{V}_\sigma = \left[L^2(\Omega)\right]^6$ is the space where the stress field is being sought.

In contrast, the identification of material properties is often referred to as an inverse problem, i.e., determining $\mathscr{A}$ with the knowledge of boundary conditions and supplementary information. In this article, such additional information is the measurements $\tilde{\boldsymbol{u}}$ of the displacement field. In practice, the value of $\tilde{\boldsymbol{u}}$ can be obtained by full-field measurements with digital video equipments and associated image correlation processes. Then, we can differentiate the displacement field to compute the strain components.

3. Methodology

3.1. General concepts of PINNs

The idea of PINNs is to embed physical knowledge into the loss function of a neural network so that the training process is restricted to a specific space. PINNs are based on two properties of the neural network: 1) the neural network is a universal approximator that can be used to approximate any function; 2) the derivative of the output of a neural network with respect to any input can be easily obtained using AD.

To illustrate the concept of PINNs, we consider the following parametrized PDE system:

$$\begin{aligned} &f\left(\boldsymbol{x}; \tfrac{\partial p}{\partial x_1}, \ldots, \tfrac{\partial p}{\partial x_d}; \tfrac{\partial^2 p}{\partial x_1^2}, \ldots, \tfrac{\partial^2 p}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) = 0, \quad \boldsymbol{x} \in \Omega \\ &g(p, \boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\Omega \end{aligned} \tag{6}$$

where $\boldsymbol{x} = [x_1, \ldots, x_d]$ is the spatial coordinate of d dimensions and p is the solution of the PDE; f denotes the residual which contains differential terms; $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots]$ are the PDE parameters and $g(p, \boldsymbol{x})$ is the boundary conditions. For time-dependent problems, Eq. (6) could also include time component and initial condition.

In the context of PINNs, a dedicated neural network is first introduced and used to approximate the PDE solution p. There are many types of neural networks can be applied, and we take the vanilla neural network as an example. Given the input $\boldsymbol{x}$, the output of a fully-connected feed-forward neural network with multiple hidden layers can be expressed by:

$$\begin{aligned} &\boldsymbol{a}^{[0]} = \boldsymbol{x} \\ &\boldsymbol{a}^{[l]} = h\left(\boldsymbol{w}^{[l]} \boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]}\right), \quad 1 \leqslant l < L \\ &a^{[L]} = \boldsymbol{w}^{[L]} \boldsymbol{a}^{[L-1]} + \boldsymbol{b}^{[L]} \end{aligned} \tag{7}$$

where L denotes the total number of the layers and $\boldsymbol{a}^{[l]}$ is the output of the l^{th} layer; $h(\cdot)$ is the activation function; $\boldsymbol{w}^{[l]}$ and $\boldsymbol{b}^{[l]}$ represent weight matrix and bias vector, respectively.

The final output $a^{[L]}$ will be used to approximate the exact solution of the PDE. Thanks to the development of deep learning frameworks [37,38], once the forward propagation of the neural network has been configured, the derivative of the output with respect to any input can be calculated according to the computational graph using the chain rule, and the updating of model parameters is thus simpler.

After choosing the neural network, the key point is to construct the loss function based on physics information. In PINNs, the data used for training can be sorted into two groups: the collocations points $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N_f}$ for residual f and the boundary training data $\{\boldsymbol{x}^{(j)}\}_{j=1}^{N_b}$ for boundary conditions g. Let us denote the PINN approximation as $\hat{p}$, and the shared neural network parameters between these two groups can be learned by minimizing the mean squared error loss:

$$\mathscr{L} = \mathscr{L}_b + \mathscr{L}_f \tag{8}$$

where

$$\begin{aligned} \mathscr{L}_f &= \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f\left(\boldsymbol{x}^{(i)}; \frac{\partial \hat{p}}{\partial x_1}, \ldots, \frac{\partial \hat{p}}{\partial x_d}; \frac{\partial^2 \hat{p}}{\partial x_1^2}, \ldots, \frac{\partial^2 \hat{p}}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right)\right|^2 \\ \mathscr{L}_b &= \frac{1}{N_b}\sum_{j=1}^{N_b}\left|g\left(\hat{p}, \boldsymbol{x}^{(j)}\right)\right|^2 \end{aligned} \tag{9}$$

The final step is training, i.e., searching for model parameters $\boldsymbol{w}^{[l]}$ and $\boldsymbol{b}^{[l]}$ to minimize the loss $\mathscr{L}$. Generally, this step requires the application of gradient-based optimizers such as Adam [39] and L-BFGS [40]. The training is complete once the loss stops decreasing or is less than a threshold.

3.2. Identification using PINNs

When dealing with the identification problem, different physical components of the system need to be considered, such as stress, strain and constitutive parameters. Such a system is difficult to be represented with only one neural network, so several neural networks are required to represent different components. Therefore, the process of identification by PINNs can be summarized as follows:

Step 1: use several neural networks to represent different physical quantities of the system;
Step 2: collect the outputs of these neural networks and calculate their partial derivatives with AD;
Step 3: aggregate the output and its partial derivatives into a loss function and calculate the total loss;
Step 4: implement backpropagation to update the parameters of the PINN model until the loss is smaller than a threshold.

In the following sections, we propose two variants of the PINN-based identification method. Variant 1 uses the generalized Hooke's law as the principle to approximate the constitutive parameters of the material. Variant 2 performs CEGM optimization via the PINN algorithm, which essentially evaluates the error quantified by means of an energy norm.

3.2.1. Variant 1: identification with generalized Hooke equation

Considering the system mentioned in Section 2, we use three separate networks to represent the displacement, stress and constitutive parameters. For each neural network, the input is the coordinate of displacement measurements $\boldsymbol{x}$, and the number of outputs depends on the problem's dimensionality. In this article, we use $\hat{\bullet}$ to indicate the neural network's approximation, and the parameters to be identified are Young's modulus E and Poisson's ratio ν.

Next, we construct the loss function based on the physical relationships of the outputs. According to the governing equations in Section 2, the total loss can be written as:

$$\mathscr{L} = \mathscr{L}_{data} + \mathscr{L}_e + \mathscr{L}_b + \mathscr{L}_c \tag{10}$$

where

$$\begin{aligned} \mathscr{L}_{data} &= \frac{1}{N_{data}}\sum_{i=1}^{N_{data}}|\tilde{\boldsymbol{u}}(\boldsymbol{x}^{(i)}) - \hat{\boldsymbol{u}}(\boldsymbol{x}^{(i)})|^2 \\ \mathscr{L}_e &= \frac{1}{N_f}\sum_{j=1}^{N_f}|\mathrm{div}\ \hat{\boldsymbol{\sigma}}(\boldsymbol{x}^{(j)})|^2 \\ \mathscr{L}_b &= \frac{1}{N_b}\sum_{q=1}^{N_b}|\hat{\boldsymbol{\sigma}}(\boldsymbol{x}^{(q)})\cdot\boldsymbol{n} - \bar{\boldsymbol{T}}|^2 \\ \mathscr{L}_c &= \frac{1}{N_f}\sum_{j=1}^{N_f}|\hat{\boldsymbol{\sigma}}(\boldsymbol{x}^{(j)}) - \hat{\mathscr{A}}(\boldsymbol{x}^{(j)}) : \hat{\boldsymbol{\varepsilon}}(\boldsymbol{x}^{(j)})|^2 \end{aligned} \tag{11}$$

Note that $\mathscr{L}_{data}$ is the loss between displacement measurements and neural network approximation; $\mathscr{L}_e$ denotes the equilibrium loss respect to the first function of Eq. (1) and $\mathscr{L}_b$ denotes the boundary loss respect to the second function of Eq. (1); $\mathscr{L}_c$ is the constitutive loss corresponds to Eq. (3). The training set includes N_{data} measurements, N_f collocations points and N_b boundary points.

We also note that we do not use a dedicated neural network to represent the strain (corresponding to Eq. (2)) because, in elasticity, the strain in any direction can be calculated by the partial derivative of the displacement. With deep learning frameworks, the approximate strain can be easily computed with AD and applied in $\mathscr{L}_c$.

After configuring the structure of the model and the loss equations, the last step is to train the model and thus identify the system's constitutive parameters.

3.2.2. Variant 2: identification with CEGM

The CEGM measures the distance between an admissible stress field $\boldsymbol{\tau}$ and another stress field computed from a constitutive model related to an admissible displacement $\boldsymbol{v}$. Under the assumption of linear elastic, the constitutive equation gap between $\boldsymbol{\tau}$ and $\boldsymbol{v}$ is described by:

$$\mathscr{E}(\boldsymbol{v}, \boldsymbol{\tau}, \mathscr{A}) = \frac{1}{2}\int_\Omega (\boldsymbol{\tau} - \mathscr{A} : \boldsymbol{\varepsilon}[\boldsymbol{v}]) : \mathscr{A}^{-1} : (\boldsymbol{\tau} - \mathscr{A} : \boldsymbol{\varepsilon}[\boldsymbol{v}])\ \mathrm{dV} \tag{12}$$

For the identification problem, $\mathscr{A}$ is unknown, the admissible fields $\mathscr{C}$ and $\mathscr{S}$ can be altered to include all experimental information related to displacements and stresses. Then, the Hooke's operator $\mathscr{A}$ can be obtained by minimizing the constitutive equation gap:

$$\mathscr{A} = \underset{\mathscr{A}^\star \in \mathbb{A}}{\arg\min}\, J(\mathscr{A}^\star) \quad \text{with} \quad J(\mathscr{A}^\star) = \min_{(\boldsymbol{v},\boldsymbol{\tau})\in\mathscr{C}\times\mathscr{S}} \mathscr{E}(\boldsymbol{v}, \boldsymbol{\tau}, \mathscr{A}^\star) \tag{13}$$

where $\mathbb{A}$ is the set of admissible elasticity fields. To be more specific, given the displacement measurement $\tilde{\boldsymbol{u}}$, Eq. (13) becomes:

$$\mathscr{A} = \underset{\mathscr{A}^\star \in \mathbb{A}}{\arg\min}\, J(\mathscr{A}^\star) \quad \text{with} \quad J(\mathscr{A}^\star) = \min_{\boldsymbol{\tau}\in\mathscr{S}} \mathscr{E}(\tilde{\boldsymbol{u}}, \boldsymbol{\tau}, \mathscr{A}^\star) \tag{14}$$

Based on the above principles, a new variant of the identification method can be derived by combining the CEGM knowledge with PINNs. Here we give another function called CEGM loss $\mathscr{L}_{CEGM}$, and the total loss becomes:

$$\mathscr{L} = \mathscr{L}_{data} + \mathscr{L}_e + \mathscr{L}_b + \mathscr{L}_{CEGM} \tag{15}$$

where

$$\mathscr{L}_{CEGM} = \frac{1}{N_f}\sum_{j=1}^{N_f}\left|\left(\hat{\boldsymbol{\sigma}}(\boldsymbol{x}^{(j)}) - \hat{\mathscr{A}}(\boldsymbol{x}^{(j)}) : \hat{\boldsymbol{\varepsilon}}(\boldsymbol{x}^{(j)})\right) : \hat{\mathscr{A}}(\boldsymbol{x}^{(j)})^{-1} : \left(\hat{\boldsymbol{\sigma}}(\boldsymbol{x}^{(j)}) - \hat{\mathscr{A}}(\boldsymbol{x}^{(j)}) : \hat{\boldsymbol{\varepsilon}}(\boldsymbol{x}^{(j)})\right)\right|^2 \tag{16}$$

Fig. 1 shows the forward propagation of the identification method. The input $\boldsymbol{x} = [x_1, \dots x_d]$ are coordinate points, including measuring coordinates, collocations points, and boundary coordinates. These coordinates points will be transmitted to different neural networks to approximate different quantities and then calculate the loss. In each iteration, the backward propagation will start with the total loss $\mathscr{L}$ and update the parameters of three models simultaneously. Details about the implementation of the developed method will be elaborated in the next section.

The presented work is carried out within the framework of linear elasticity. The method is of particular interest due to its simple implementation of the constitutive relation error concept. The primary step involves building stress, which verifies the equilibrium equations. The constitutive relation error is then defined as a cost function. The proposed method minimizes both the non-admissibility of the admissible stress and the cost function simultaneously. In contrast, the classical approach involves performing the two steps separately, resulting in two loops being set up: the first for building an admissible field and the second for minimizing the constitutive error.

Generally speaking, Variant 2 is based on the same principle as CEGM to minimize constitutive error, yet in terms of underlying assumptions, they differ from each other in two aspects. Firstly, from a theoretical point of view, an admissible stress field must be built at all points of the domain in the CEGM, which is the main difficulty of this method. By contrast, the equilibrium is only enforced among several selected points (which could be considerable) in Variant 2, making the stresses in equilibrium at many points. Secondly, in the CEGM, the tangential stresses are chosen discontinuously between two subregions, and Young's modulus is considered a constant at each subregion. In the proposed method, the constitutive parameters and admissible stress are considered as continuous functions of position $\boldsymbol{x}$: $\widehat{E}(\boldsymbol{x}), \hat{\nu}(\boldsymbol{x})$ and $\hat{\boldsymbol{\sigma}}(\boldsymbol{x})$. Thus, the developed method has a wider range of applications because there is no limitation on the parameter form.

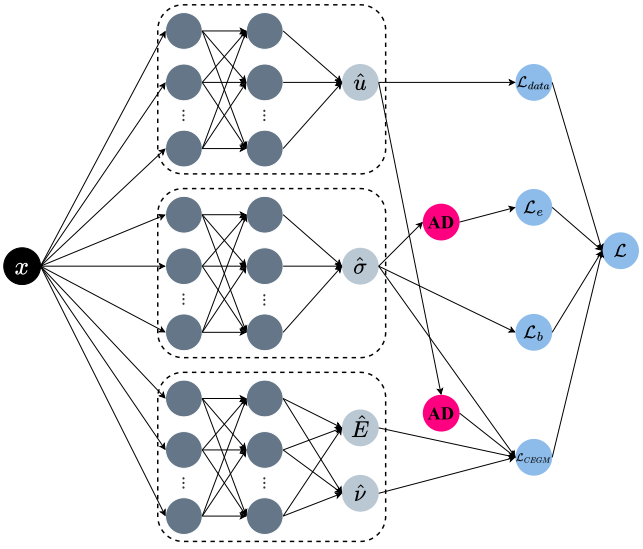

Fig. 1. Forward propagation of identification using PINNs.

Regarding the difference between the two variants, it is basically the different weights for the different constitutive equations. Variant 1 assigns identical weights to each part of the constitutive relation, whereas Variant 2 naturally provides weights corresponding to the constitutive relation error, which is generally a more optimal approach. Additionally, Variant 2 incorporates the concept of the constitutive relation error, enabling the method to handle non-linear material behavior. Variant 1 represents an explicit writing of the constitutive law, while Variant 2 is a more general approach that relies on the decomposition into the admissible field and measuring the constitutive relation error. While Variant 1 seems difficult to extend to more complex behavior, Variant 2 offers a more versatile framework that can be extended to more complex behaviors, thanks to the constitutive relation error concept.

Remark 1. In this study, we present a general approach for utilizing the proposed method to solve more complex nonlinear problems within the framework of Variant 2. Building on the dissipation concept introduced by Ladeveze, the following steps are necessary: 1) introducing and constructing new fields in addition to stress; 2) defining the constitutive error with respect to the constitutive law. For instance, in [41] the elastoplasticity framework is developed, wherein a detailed description of the new set of admissible internal variables is provided. Notably, in this framework, the proposed method can directly construct the set of new admissible internal variables. The dissipation error serves as the cost function to be minimized by the PINN.

By incorporating the dissipation concept, the proposed method offers a versatile framework for addressing complex nonlinear problems in mechanics. The construction of new fields and the definition of the constitutive error provide additional degrees of freedom for capturing the nonlinear behavior of materials. The approach is particularly useful in the context of elastoplasticity, where the PINN can effectively learn the plastic behavior of materials based on the new set of internal variables.

Further research is necessary to evaluate the performance of this approach on a wider range of nonlinear problems and to explore potential extensions to other fields of mechanics. Nonetheless, the findings of this study suggest that the proposed method holds promise as a powerful tool for tackling complex nonlinear problems in mechanics.

4. Numerical examples

4.1. Problem setup

To better evaluate the performance of the proposed method, here we cite examples from the authors' previous paper and compare the results obtained by different approaches. Consider a 10 mm × 10 mm square in the x-y plane. Two loadings and two configurations of the material heterogeneities make up three reference problems (RP1 - RP3), illustrated in Fig. 2.

The square can be divided into $N_s = 100$ zones of 1 mm × 1 mm, which will be used subsequently for evaluation. All three problems were solved using *Cast3M* software (www-cast3m.cea.fr) with a fine mesh. Then 121 samples based on a uniformly distributed coarse grid (see Fig. 3) were extracted as displacement measurements. The goal is to identify the material properties with these measurements. More details are illustrated in [9].

4.2. Technical details

In order to identify the constitutive parameters of the reference problems, the first step is to determine how to implement the for-

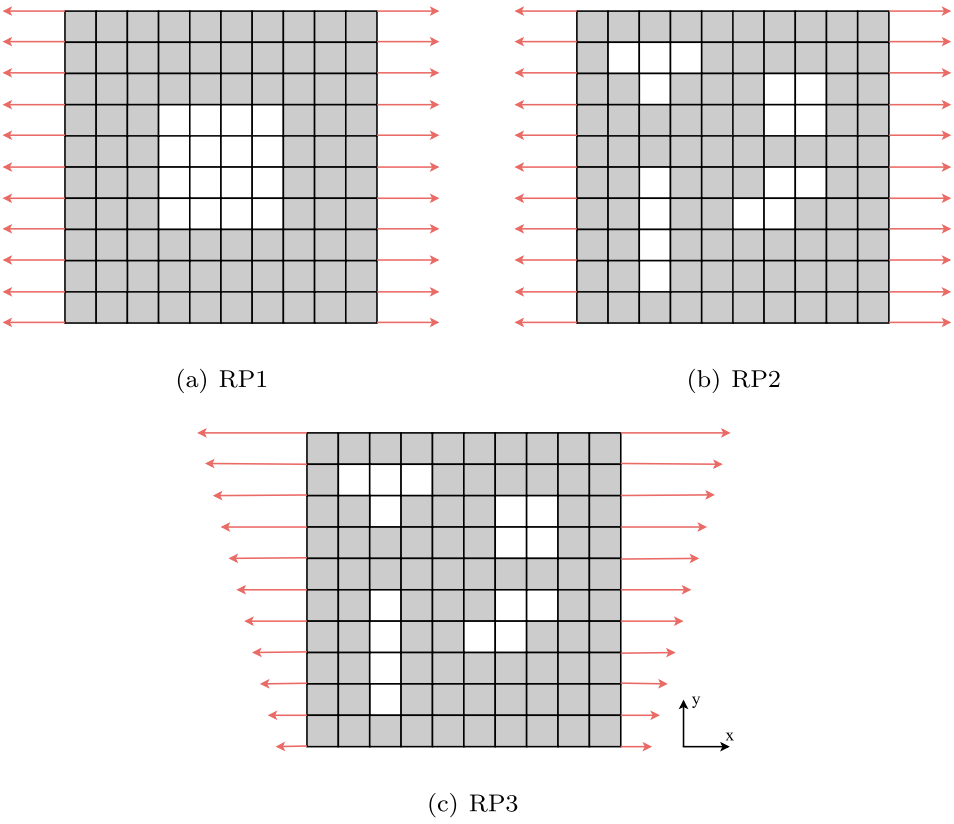

Fig. 2. Maps of material heterogeneities and their loading. (a) White zones: $E = 2\,\text{MPa}$ and $\nu = 0.3$; grey zones: $E = 1\,\text{MPa}$ and $\nu = 0.3$; loading: $\boldsymbol{\sigma}_{xx} = 1\,\text{MPa}$. (b) White zones: $E = 2\,\text{MPa}$ and $\nu = 0.3$; grey zones: $E = 1\,\text{MPa}$ and $\nu = 0.3$; loading: $\boldsymbol{\sigma}_{xx} = 1\,\text{MPa}$. (c) White zones: $E = 2\,\text{MPa}$ and $\nu = 0.3$; grey zones: $E = 1\,\text{MPa}$ and $\nu = 0.3$; loading: $\boldsymbol{\sigma}_{xx} = (0.1y + 0.5)\,\text{MPa}$.

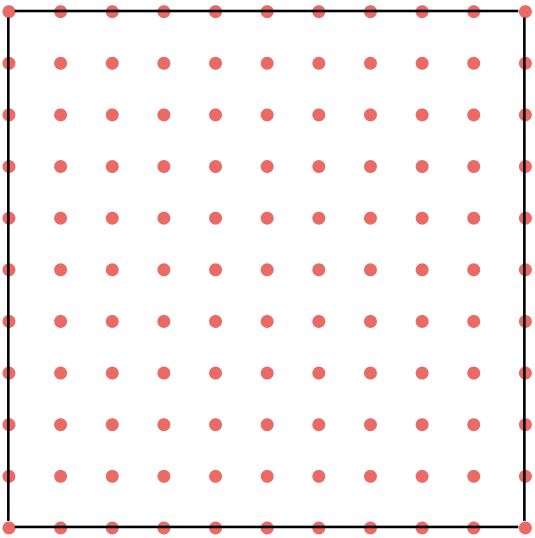

Fig. 3. Coarse grid (11×11) for displacement measurements.

ward propagation of PINN. In fact, the propagating ways is varied and flexible. In this work, we apply the architecture shown in Fig. 1 for all experiments with the help of *PyTorch* [37] framework. Nevertheless, according to the authors' experiments, it is also feasible to represent physical components in each direction with different neural networks, for instance, approximating $\hat{u}_x$ and $\hat{u}_y$ with two separate networks. However, introducing more sub-networks also increases the number of neural network parameters (weights and biases) that must be backpropagated. In order to illustrate the method more explicitly, we used an intuitive neural network structure, i.e., with three networks representing displacement, stress, and constitutive parameters, respectively. For comparison purposes, we use identical hyperparameters for both variants to cope with the three reference problems. For RP1-RP3, all three vanilla neural networks contained in the PINN have 3 hidden layers with 32 neurons for each layer. The collocation points are 400 (20×20) x-y coordinates uniformly distributed over the entire plane. All samples used to calculate the boundary loss come from collocation points distributed at the boundary. The *tanh* function is selected as the default activation function. The optimizer used is Adam, and the learning rate is fixed at 0.0001. The PINN models are trained 40,000 epochs for each problem.

Remark 2. Like other deep learning techniques, we need to tune hyperparameters to achieve good accuracy. When addressing the problems in this paper, the hyperparameters mainly include learning rate, network size, and the number of collocation points. For PINNs, the choice of network size is highly dependent on the complexity of the PDE solution. Even for single-dimensional problems, the required depth of networks to achieve similar levels of accuracy can vary greatly. For identification using PINNs, the results are effected by each sub-network. For example, when we choose a small network size to approximate a very complicated constitutive parameter distribution, even if the PINN model learns the measurement information well, it is still difficult to identify correctly. We can avoid these problems by monitoring the loss and tuning the hyperparameters accordingly. In addition, we found

that PINNs do not require too many collocation points in identification problems. Increasing the number of collocation points does not help when the model reaches a certain accuracy.

Moreover, unlike traditional numerical methods that have been proven to be convex, the loss of PINNs is highly nonlinear and nonconvex, which results in no guarantee of unique solutions. We also found that the identified parameters will converge to different values from different initial weights of network. As Lu Lu et al. said [21], a useful strategy is to train PINNs several times from random initialization and adopt the model with the smallest loss as the final solution. In this work, our focus is not to get the best models but to illustrate ideas, and better results can be obtained by tuning hyperparameters.

4.3. Results

To evaluate the proposed method, we compare the identified parameters with their exact values in each zone (1 mm × 1 mm square in Fig. 2). First, we indicate the the relative error e_r as follows:

$$e_r = \frac{|\square^{\boxplus} - \square^{exact}|}{|\square^{exact}|} \tag{17}$$

where $\square = E$ or ν, and $\boxplus =$ Variant 1 or Variant 2. $\square^{\boxplus}$ contains the mean values in every zone, and these values are calculated from a test set of 100 points sampled from each zone. Fig. 4–6 show the error maps of two variants for the three reference problems, respectively. It can be seen in these figures that the two variants achieve similar performance on the three problems. Unlike the CEGM assumes that the material properties of each zone remain constant, the proposed method tries to approximate discontinuous Young's modulus distribution with a continuous function, which can result in larger errors at the point of the junction where the material properties change.

Then, for global analysis, we calculate the mean, maximum and standard deviation of e_r. Table 1–3 show the results of the PINN-related methods and two classical identification strategies. The data on EGM and CEGM is also quoted from the authors' previous paper. The citation of these data allows us to compare the quality of the proposed method with the commonly used EGM and the classical CEGM. We can observe that overall the PINN models achieve a similar level of accuracy as the classical finite element method. Specifically, the proposed method performs close to EGM but not as well as CEGM in identifying E; in terms of ν, it gives results as good as CEGM, and both methods outperform EGM. Besides that, Variant 2 performs slightly better than Variant 1.

The time cost comparison between the two variants and CEGM is shown in Table 4. To identify the parameters of each reference problem, the proposed method took an average of 11 min on the python platform (running on CPU), while CEGM took more than 4 h on the matlab platform. Even though based on different platforms, the difference in computational efficiency comes from the algorithms themselves. The CEGM requires a double loop to optimize the admissible stress and to minimize the function. The double loop can be very greedy if the required equilibrium is verified precisely. In the developed method, the stress is built by PINN in

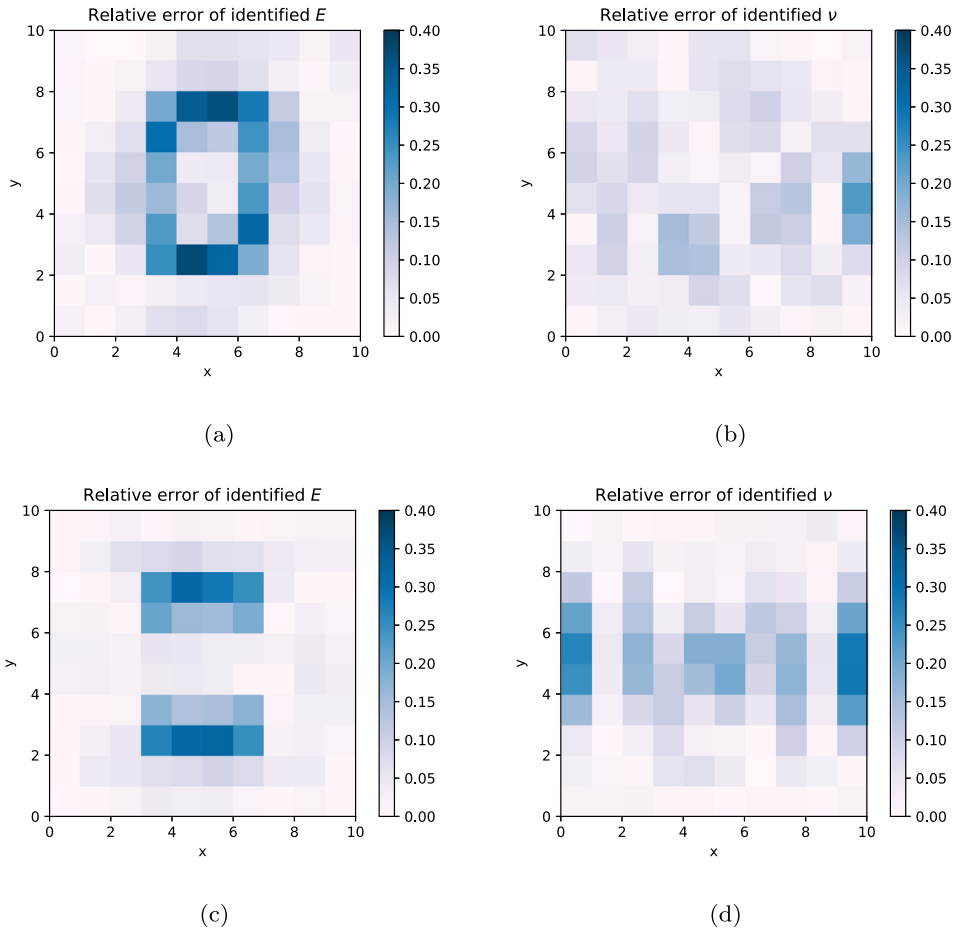

Fig. 4. Comparison of e_r between two variants for RP1. (a), (b): Variant 1; (c), (d): Variant 2.

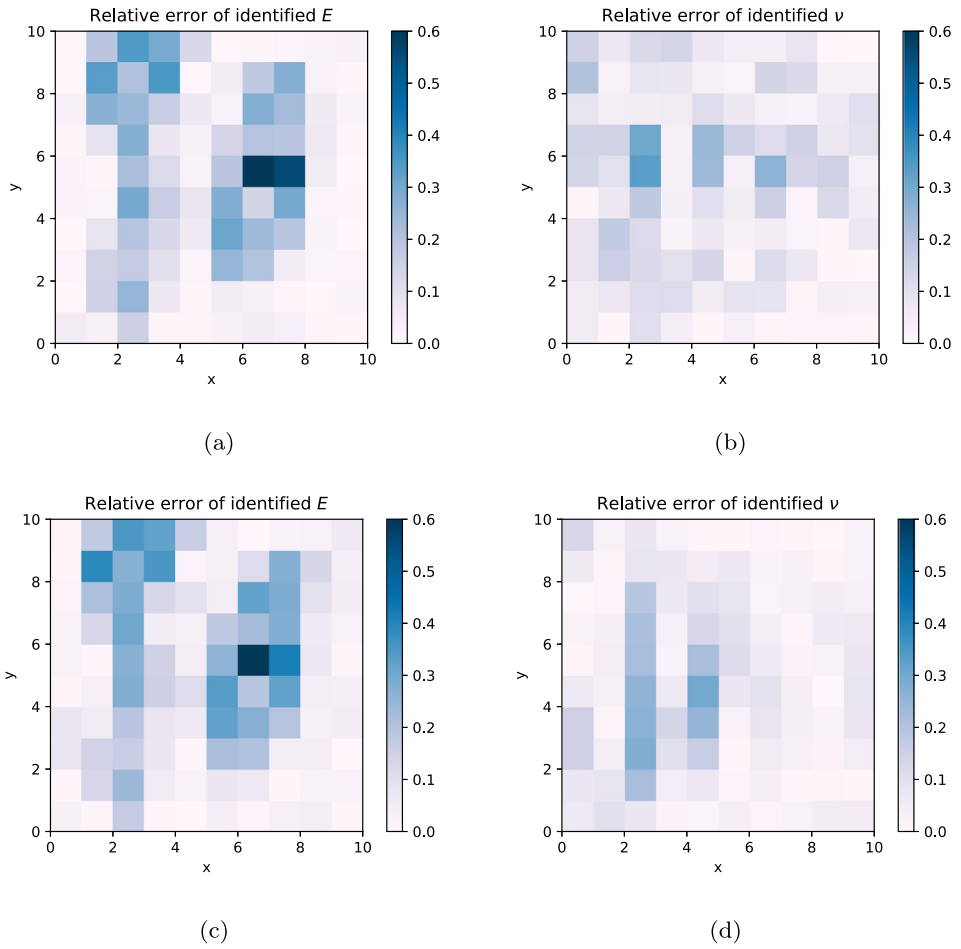

Fig. 5. Comparison of e_r between two variants for RP2. (a), (b): Variant 1; (c), (d): Variant 2.

the training process. Consequently, less attention is paid to constructing admissible fields, and a comparable accuracy is achieved.

5. Application to 3D problem

In this section, we investigate the effectiveness of the proposed method on different 3D problems. When using CEGM to tackle 3D problems, one has to build the admissible stress field in 3D, which seems uncomplicated in theory but turns out to be quite complex in practice [42]. However, such an extension is much simpler under the proposed method. The only thing that needs to be done is to add the physical information of the z direction to the PINN, which just adds a dimension to the set of the admissible equations.

5.1. 3D Example 1

We consider a cube shown in Fig. 7. The size of the cube is $10\,\text{mm} \times 10\,\text{mm} \times 10\,\text{mm}$, and it can be divided into 1000 zones of $1\,\text{mm} \times 1\,\text{mm} \times 1\,\text{mm}$ which will be used to test the PINN model. The core of this cube and the rest part have different material properties, and we apply a y-axis-dependent force on its sides. The problem is solved with a fine mesh, and the displacements of the vertices of each subset (1331 sampling points total) are extracted as training data for PINN.

Here we apply the Variant 2 of the proposed method. Similar to the 2D problems, we use three vanilla neural networks to represent different physical quantities, and all of these networks have 4 hidden layers with 32 neurons in each layer. Other hyperparameters include the usage of Adam optimizer, 68921 ($41 \times 41 \times 41$) collocation points, and 0.0001 learning rate. After training about 25,000 epochs, the PINN model is evaluated by uniformly selecting 1000 sampling points in each $1\,\text{mm} \times 1\,\text{mm} \times 1\,\text{mm}$ zone, calculating the output and taking the average value.

To demonstrate the difference between the identified parameters and their exact values, we define the following quantity:

$$\text{Ratio} = \frac{\square^{\boxplus}}{\square^{exact}} \tag{18}$$

Fig. 8 shows the Ratio of two parameters. The abscissa is the serial number of the 1000 zones, and the ordinate is the computed ratio. The red line with a ratio equal to 1 represents the best possible approximations. Compared to the exact values, the identified Young's modulus E and Poisson's ratio ν have mean relative errors of 0.08 and 0.04, respectively. Although the problem's dimensionality has increased, the developed method is still effective in identifying these two parameters.

5.2. 3D Example 2

In this section, we build another 3D model that has an analytical solution with smooth variations over the domain. With this example, we analyze the convergence of the proposed method with different collocation point numbers and evaluate its robustness. Variant 2 is applied as the default method to solve this problem.

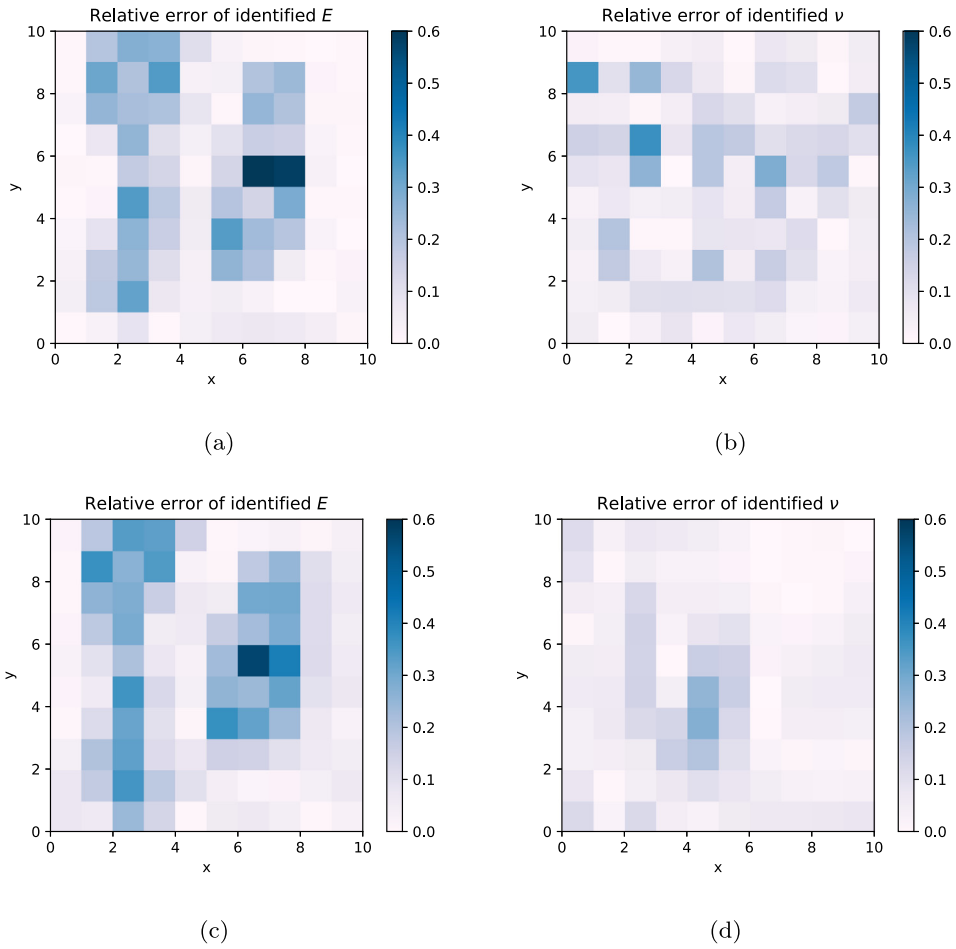

Fig. 6. Comparison of e_r between two variants for RP3. (a), (b): Variant 1; (c), (d): Variant 2.

Table 1
Mean error $\left(\frac{\sum e_r}{N_s}\right)$ comparison among four different methods.

	E[RP1]	ν[RP1]	E[RP2]	ν[RP2]	E[RP3]	ν[RP3]
EGM	0.06	0.14	0.10	0.12	0.09	0.12
CEGM	0.02	0.04	0.05	0.07	0.05	0.07
Variant 1	0.08	0.06	0.13	0.07	0.12	0.08
Variant 2	0.06	0.07	0.10	0.09	0.11	0.09

Table 2
Maximum error (max e_r) comparison among four different methods.

	E[RP1]	ν[RP1]	E[RP2]	ν[RP2]	E[RP3]	ν[RP3]
EGM	0.27	0.59	0.68	0.54	0.56	0.40
CEGM	0.13	0.19	0.43	0.30	0.42	0.30
Variant 1	0.37	0.23	0.74	0.25	0.73	0.38
Variant 2	0.32	0.29	0.61	0.36	0.67	0.34

Table 3
Standard deviation of error (std e_r) comparison among four different methods.

	E[RP1]	ν[RP1]	E[RP2]	ν[RP2]	E[RP3]	ν[RP3]
EGM	0.07	0.15	0.12	0.11	0.10	0.10
CEGM	0.04	0.05	0.08	0.07	0.08	0.07
Variant 1	0.09	0.04	0.13	0.06	0.13	0.07
Variant 2	0.08	0.07	0.11	0.08	0.12	0.08

Table 4
Time cost comparison among the proposed method and the CEGM.

	Variant 1	Variant 2	CEGM
RP1	8 min 10 s	8 min 18 s	> 4 h
RP2	24 min 1 s	8 min 24 s	
RP3	8 min 9 s	8 min 19 s	

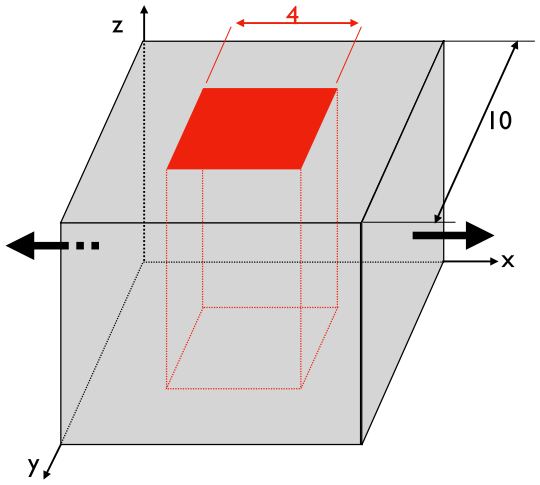

Fig. 7. Configuration of 3D Example 1. The core of the cube (red zones) which contains 160 zones: $E = 2\,\text{MPa}$ and $\nu = 0.3$; The rest part (grey zones): $E = 1\,\text{MPa}$ and $\nu = 0.3$; loading: $\boldsymbol{\sigma}_{xx} = (0.1y + 0.5)\,\text{MPa}$.

Consider a cube in 3D space. The side length of the cube is π mm, i.e., the three coordinates x, y and z belong to $[0, \pi]$. The constitutive parameters of the cube material are functions as follows:

$$\begin{cases} E = E_0(1 + sin(y)) \\ \nu = \nu_0 \end{cases} \tag{19}$$

where E_0, ν_0 are set to 1 MPa and 0.3, respectively. The loading consists of four forces applied on four faces of the cube:

- On face $x = 0 : \sigma_{xx} = 1 + sin(y)\,\text{MPa}$, $\sigma_{xy} = 0\,\text{MPa}$ and $\sigma_{xz} = 0\,\text{MPa}$
- On face $x = \pi : \sigma_{xx} = 1 + sin(y)\,\text{MPa}$, $\sigma_{xy} = 0\,\text{MPa}$ and $\sigma_{xz} = 0\,\text{MPa}$
- On face $y = 0 : \sigma_{yy} = 1\,\text{MPa}, \sigma_{xy} = 0\,\text{MPa}$ and $\sigma_{yz} = 0\,\text{MPa}$
- On face $y = \pi : \sigma_{yy} = 1\,\text{MPa}, \sigma_{xy} = 0\,\text{MPa}$ and $\sigma_{yz} = 0\,\text{MPa}$

The others faces are free-force boundaries. This test corresponds to a biaxial test in the direction x and y. According to the governing equations in Section 2, we can deduce the analytical solution of displacement, which is then used as measurements $\tilde{\boldsymbol{u}}$.

As in the 3D Example 1, we generated 1331 samples as measurements. Regarding the architecture of PINN, each sub-network has 3 hidden layers with 32 units in each layer, and the learning rate is 0.001. To investigate the effect of collocation point numbers on convergence, we train PINN models with varying numbers of points (10,000 epochs per test). Fig. 9 shows the mean relative error under different numbers of collocation points. It can be seen that the proposed method does not require many collocation points to converge, which is different from the conventional belief that more collocation points lead to lower errors. In this example, given 1331 measurements, a hundred collocation points are enough to achieve convergence for the two parameters.

In addition, we examine the robustness of the method in the presence of measurement noise. We set the number of collocation points to 1000, and the rest of the setup remains unchanged. By adding different percentages of random noise to the measurement $\tilde{\boldsymbol{u}}$, we obtained the results shown in Table 5 and 6. It can be seen

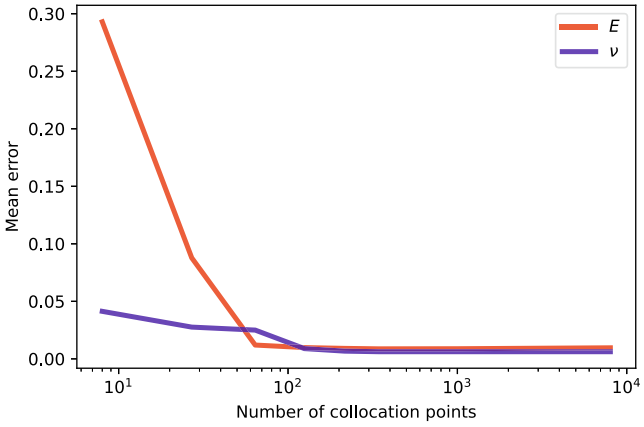

Fig. 9. Mean error $\left(\frac{\sum e_r}{N_s}\right)$ of identified parameters with different number of collocation points.

Table 5
Error analysis under different percentage of noise for Young's modulus E.

Noise	mean e_r	max e_r	std e_r
0%	0.0088	0.0715	0.0086
1%	0.0086	0.0704	0.0085
5%	0.0086	0.0722	0.0082
10%	0.0106	0.1217	0.0141

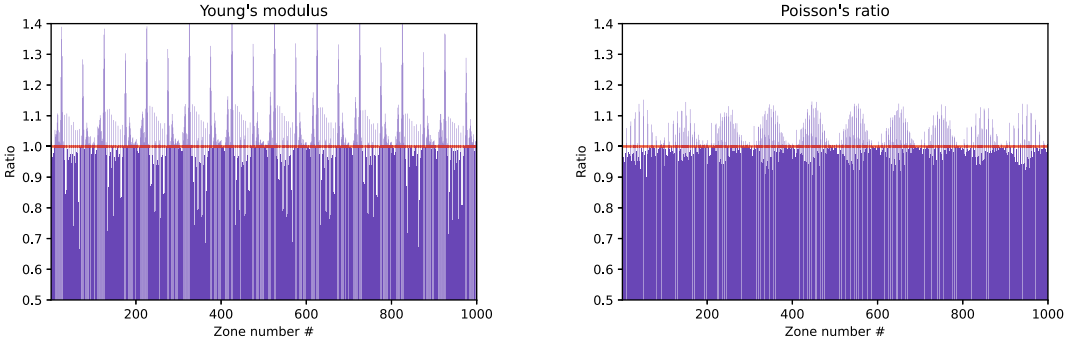

Fig. 8. Ratio comparison for the 3D Example 1.

Table 6
Error analysis under different percentage of noise for Poisson's ratio v.

Noise	mean e_r	max e_r	std e_r
0%	0.0062	0.0475	0.0051
1%	0.0063	0.0464	0.0052
5%	0.0076	0.0490	0.0059
10%	0.0120	0.0555	0.0100

that the measurement noise within 5% has little influence on the results in this example. As the noise continues to increase, the accuracy tends to decrease.

6. Conclusion

We presented a method to identify the constitutive parameters of heterogeneous linear elastic materials using PINNs. The proposed method was based on the CEGM principle and performed the optimized process via the PINN algorithm. Compared with classical identification strategies, the new approach avoided the laborious implementation procedure and offered higher computation efficiency. The introduction of constitutive relation error concept also made it possible to apply the method to nonlinear material behavior. We tested the method with three 2D reference problems. The results showed that the developed method achieved the same level of identification accuracy as classical strategies, even when the discontinuity problem setup was more favorable to the latter. We also illustrated the effectiveness of the method through two different 3D problems, one with discontinuous parameter variations and the other with continuous variations. The developed tool was an attempt to combine traditional theories with new techniques, which hopefully will provide novel ideas for industrial applications. Extending the method to dynamical or nonlinear behavior is an interesting direction for future work. Moreover, we adopted a very basic neural network in this first work, and it is also worth exploring in depth using more complex networks and other optimization algorithms.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: [WEI reports financial support was provided by China Scholarship Council.]

Acknowledgement

Y.W. thanks the China Scholarship Council (CSC) for his PhD fellowship.

References

[1] Sutton MA, vision-based Computer. noncontacting deformation measurements in mechanics: a generational transformation. Appl Mech Rev 2013;65(5).
[2] Grediac M, Sur F, Blaysat B. The grid method for in-plane displacement and strain measurement: A review and analysis. Strain 2016;52(3):205–43.
[3] Avril S, Bonnet M, Bretelle A-S, Grédiac M, Hild F, Ienny P, Latourte F, Lemosse D, Pagano S, Pagnacco E, et al. Overview of identification methods of mechanical parameters based on full-field measurements. Exp Mech 2008;48(4):381–402.
[4] Kavanagh KT, Clough RW. Finite element applications in the characterization of elastic solids. Int J Solids Struct 1971;7(1):11–23.
[5] Claire D, Hild F, Roux S. Identification of damage fields using kinematic measurements. Comptes Rendus Mécanique 2002;330(11):729–34.
[6] Claire D, Hild F, Roux S. A finite element formulation to identify damage fields: the equilibrium gap method. Int J Num Methods Eng 2004;61(2):189–208.
[7] Grédiac M. Principe des travaux virtuels et identification, Comptes rendus de l'Académie des sciences. Série 2, Mécanique. Phys Chim Sci de l'univers, Sci de la Terre 1989;309(1):1–5.
[8] Ladeveze P, Leguillon D. Error estimate procedure in the finite element method and applications. SIAM J Num Anal 1983;20(3):485–509.
[9] Florentin E, Lubineau G. Identification of the parameters of an elastic material model using the constitutive equation gap method. Comput Mech 2010;46(4):521–31.
[10] Florentin E, Lubineau G. Using constitutive equation gap method for identification of elastic material parameters: technical insights and illustrations. Int J Interact Des Manuf (IJIDeM) 2011;5(4):227–34.
[11] Martins J, Andrade-Campos A, Thuillier S. Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements. Int J Mech Sci 2018;145:330–45.
[12] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[13] Li W, Bazant MZ, Zhu J. A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. Comput Methods Appl Mech Eng 2021;383:113933.
[14] Yu B et al. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun Mathe Stat 2018;6(1):1–12.
[15] Sirignano J, Spiliopoulos K. Dgm: A deep learning algorithm for solving partial differential equations. J Comput Phys 2018;375:1339–64.
[16] Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of kirchhoff plate, arXiv preprint arXiv:2102.02617.
[17] Nguyen-Thanh VM, Zhuang X, Rabczuk T. A deep energy method for finite deformation hyperelasticity. Eur J Mech-A/Solids 2020;80:103874.
[18] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[19] Poggio T, Mhaskar H, Rosasco L, Miranda B, Liao Q. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. Int J Autom Comput 2017;14(5):503–19.
[20] Grohs P, Hornung F, Jentzen A, Von Wurstemberger P. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations, arXiv preprint arXiv:1809.02362.
[21] Lu L, Meng X, Mao Z, Karniadakis GE. Deepxde: A deep learning library for solving differential equations. SIAM Rev 2021;63(1):208–28.
[22] Pang G, Lu L, Karniadakis GE. fpinns: Fractional physics-informed neural networks. SIAM J Sci Comput 2019;41(4):A2603–26.
[23] Yang L, Zhang D, Karniadakis GE. Physics-informed generative adversarial networks for stochastic differential equations. SIAM J Sci Comput 2020;42(1):A292–317.
[24] Zhang D, Lu L, Guo L, Karniadakis GE. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. J Comput Phys 2019;397:108850.
[25] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science 2020;367(6481):1026–30.
[26] Zhu Q, Liu Z, Yan J. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. Comput Mech 2021;67(2):619–35.
[27] Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE. Physics-informed neural networks (pinns) for fluid mechanics: A review. Acta Mech Sin 2021;37(12):1727–38.
[28] Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A deep learning framework for solution and discovery in solid mechanics, arXiv preprint arXiv:2003.02751.
[29] Zhang E, Dao M, Karniadakis GE, Suresh S. Analyses of internal structures and defects in materials using physics-informed neural networks. Sci Adv 2022;8(7):eabk0644.
[30] Shukla K, Di Leoni PC, Blackshire J, Sparkman D, Karniadakis GE. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. J Nondestr Eval 2020;39:1–20.
[31] Yan C, Vescovini R, Dozio L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. Comput Struct 2022;265:106761.
[32] Tartakovsky AM, Marrero CO, Perdikaris P, Tartakovsky GD, Barajas-Solano D. Learning parameters and constitutive relationships with physics informed deep neural networks, arXiv preprint arXiv:1808.03398.
[33] Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput Methods Appl Mech Eng 2021;379:113741.
[34] Zhang E, Yin M, Karniadakis GE. Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging, arXiv preprint arXiv:2009.04525.
[35] Anton D, Wessels H. Identification of material parameters from full-field displacement data using physics-informed neural networks, Researchgate Preprint.
[36] Kamali A, Sarabian M, Laksari K. Elasticity imaging using physics-informed neural networks: Spatial discovery of elastic modulus and poisson's ratio. Acta Biomater 2023;155:400–9.
[37] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in pytorch. In: NISP 2017 Workshop Autodiff; 2017.

[38] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467.
[39] Kingma DP, Ba J. Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[40] Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 1995;16(5):1190–208.
[41] Blaysat B, Florentin E, Lubineau G, Moussawi A. A dissipation gap method for full-field measurement-based identification of elasto-plastic material parameters. Int J Num Methods Eng 2012;91(7):685–704.
[42] Florentin E, Guinard S, Pasquet P. A simple estimator for stress errors dedicated to large elastic finite element simulations: Locally reinforced stress construction. Eng Comput 2011;28(1):76–92.