Probabilistic Engineering Mechanics 68 (2022) 103240

Contents lists available at ScienceDirect

Probabilistic Engineering Mechanics

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

# Interval and fuzzy physics-informed neural networks for uncertain fields

Jan N. Fuhg [a,*], Ioannis Kalogeris [b], Amélie Fau [c], Nikolaos Bouklas [a,d]

[a] Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, United States
[b] School of Civil Engineering. National Technical University of Athens, Greece
[c] Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT, Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France
[d] Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, United States

ARTICLE INFO

*Keywords:*
Physics-informed machine learning
Fuzzy set theory
Interval set theory
Non-probabilistic uncertainty

ABSTRACT

Temporally and spatially dependent uncertain parameters are regularly encountered in engineering applications. Commonly these uncertainties are accounted for using random fields and processes, which require knowledge about the appearing probability distributions functions that is not readily available. In these cases non-probabilistic approaches such as interval analysis and fuzzy set theory are helpful to analyze uncertainty. Partial differential equations involving fuzzy and interval fields are traditionally solved using the finite element method where the input fields are sampled using some basis function expansion methods. This approach however relies on information about the spatial correlation of the fields, which is not always obtainable. In this work we utilize physics-informed neural networks (PINNs) to solve interval and fuzzy partial differential equations. The resulting network structures termed interval physics-informed neural networks (iPINNs) and fuzzy physics-informed neural networks (fPINNs) show promising results for obtaining bounded solutions of equations involving spatially and/or temporally uncertain parameter fields. In contrast to finite element approaches, no correlation length specification of the input fields as well as no Monte-Carlo simulations are necessary. In fact, information about the input interval fields is obtained directly as a byproduct of the presented solution scheme. Furthermore, all major advantages of PINNs are retained, i.e. meshfree nature of the scheme, and ease of inverse problem set-up.

## 1. Introduction

Uncertain parameters in engineering applications are commonly found to have temporal and spatial dependencies. These uncertainties include external loading, e.g. in the case of wind loads, as well as material properties e.g. as found in heterogeneous media and biological tissues. Commonly, these types of uncertainties have been modeled as stochastic or random fields [1,2]. In stochastic fields the randomness is characterized by correlated random variables at each time increment and spatial location. When random fields appear in partial differential equations (PDEs) they generalize to stochastic PDEs (sPDEs), which are commonly solved using the so-called stochastic finite element method [3,4]. In this context, the conceptually simplest approach for uncertainty propagation are Monte-Carlo simulations, which are still commonly applied as the main benchmarking approach for probabilistic methods. However they generally require a large number of samples, which typically leads to high computational costs. In order to combat this, surrogate modeling techniques are used to vastly decrease the necessary computation time [5,6].

A major bottleneck of probabilistic methods is the requirement of precise knowledge of the probability distribution functions of the uncertain parameters. This generally necessitates a great amount of statistical information in form of experimental data which is hard to come by. Furthermore, following Möller et al. [7] and Bothe [8], stochastic uncertainty can also only be used if an event, which is a random result of an experiment, can be observed under constant boundary conditions. However, if the boundary conditions are (seemingly) changing or a non-significant amount of data is available there is a crucial lack of information which is called informal uncertainty. Additionally, in certain problems, available information is only known with lexical imprecision. Informal and lexical uncertainties are commonly present in engineering science and are areas where classical probabilistic methods may only be applied to a limited extend [9].

For this reason, non-probabilistic approaches such as interval analysis [10–12] and fuzzy set theory [13–15] have been proposed. They can be used when informal or lexical uncertainties are present by describing these uncertainties through interval and fuzzy parameters. PDEs involving interval or fuzzy parameters generalize to interval and fuzzy PDEs (iPDEs and fPDEs). Similarly to sPDEs, these constructs can be solved numerically with variants of the finite element method, e.g. interval finite element analysis [16–18] and fuzzy finite element

* Corresponding author.
*E-mail address:* jf853@cornell.edu (J.N. Fuhg).

https://doi.org/10.1016/j.probengmech.2022.103240
Received 22 November 2021; Received in revised form 7 February 2022; Accepted 13 February 2022
Available online 18 February 2022

analysis [19–21]. However, interval finite element and fuzzy finite element methods, as they pertain to the analysis of spatial and time-variable uncertainty fields, are still areas of active research since they are accompanied by high computational complexity [22]. For example, efficient and convenient constructions of the interval fields are still posing a challenge [18] since they are typically constructed following some basis function expansion approach, based on the Karhunen–Loève (KL) decomposition [18,23–27]. Efficient intrusive approaches for uncertainty propagation have been developed based on these series expansions [18,28] but they require case-specific formulations to handle nonlinear problems with non-affine dependence on the interval fields. In addition, KL-like expansions require knowledge of the spatial correlation of the input fields, which might be assumed but are in general not known due to lack of experimental data and difficulties in identification [29]. In this work we present a method based on artificial intelligence that completely bypasses the issues of all existing approaches, and obtains a description of the input field as a byproduct of the solution procedure.

Machine learning technologies have been an emergent tool in computational engineering in recent years. They have been used for data-driven constitutive modeling [30–33] as a means to enable hierarchical multiscale calculations or for the development of intrusive and non-intrusive Reduced Order Modeling (ROM) schemes for accelerated solutions of PDEs [34–36]. Recently, machine learning has also been employed as a solution scheme for PDEs. This approach called physics-informed neural networks (PINNs) [37–40] has emerged as an alternative numerical technique for solving PDEs in a forward and inverse manner. Here, in an unsupervised manner, neural networks are used to approximate PDE solutions utilizing automatic differentiation to define global shape functions for the solutions. This formulation has already been extended to sPDEs with uncertain fields in [41–43]. One bottleneck with these approaches is that they still require information about the probability distribution functions which in turn requires a statistically significant amount of experimental data.

In this work we present a method for solving iPDEs and fPDEs based on machine learning that completely bypasses the issues of all existing approaches and obtains a description of the input field as a byproduct of the solution procedure. Specifically, we introduce interval physics-informed neural networks (iPINN), which are also extended here to fuzzy physics-informed neural networks (fPINN). The proposed iPINN framework consists of a dedicated network architecture and an appropriately modified loss function in order to obtain the extrema of the PDE's solution field, while at the same time satisfying the physics of the problem. Importantly, the proposed methodology can be applied to any type of problem (linear or nonlinear) and any type of interval field (homogeneous or nonhomogenous) and fuzzy fields that can be described by convex interval fields, without requiring any information about the correlation structure of these fields. Furthermore, its ease of implementation and generalization capabilities, allows it to be straightforwardly applied to any type of fuzzy and interval PDE encountered in engineering applications. In this frame, iPINN can be understood as a component of fPINN, on which we will elaborate in an upcoming section.

The paper is organized as follows. Interval and fuzzy variables and fields and PDEs involving these types of uncertainty representations are shortly reviewed in Section 2. The iPINN and fPINN formulations are introduced in Section 3, subsequently, these formulations are applied and studied through a series of numerical examples in Section 4. The paper is concluded in Section 5.

## 2. Interval and fuzzy fields and partial differential equations

Even though interval and fuzzy set analysis has seen interest since the 1960s [13], they have not received the same attention across all fields of computational physics. For this reason we use this section to describe the core concepts of fuzzy and interval numbers. An uncertain parameter $\boldsymbol{X}$ can be represented as an interval parameter $\boldsymbol{X}^I$ if it can be defined by a variation range

$$\boldsymbol{X} \in \boldsymbol{X}^I = \left[\boldsymbol{X}^L, \boldsymbol{X}^U\right] \tag{1}$$

with the lower and upper limits $\boldsymbol{X}^L$ and $\boldsymbol{X}^U$ respectively. An extension to this idea are interval fields. Consider a spatial location $\boldsymbol{x} \in D$ in a domain $D \subset \mathcal{R}^d$ and let the time be defined by $t \in \mathcal{T}$. A time-dependent parameter field $\boldsymbol{P}(\boldsymbol{x}, t)$ is a spatially uncertain interval field $\boldsymbol{P}^I(\boldsymbol{x}, t)$ if for any $\boldsymbol{x} \in D$ and $t \in \mathcal{T}$ the field values are exactly defined by a space- and time-dependent interval

$$\boldsymbol{P}(\boldsymbol{x}, t) \in \boldsymbol{P}^I(\boldsymbol{x}, t) = \left[\boldsymbol{P}^L(\boldsymbol{x}, t), \boldsymbol{P}^U(\boldsymbol{x}, t)\right]. \tag{2}$$

From the provided definitions interval numbers can be understood as generalizations of real numbers. A further generalization are fuzzy numbers which can be seen as an extension to the interval concept.

A fuzzy number $\boldsymbol{X}^F$ is composed of the pair $(\boldsymbol{X}, \mu)$, with the membership function $\mu : \boldsymbol{X} \rightarrow [0, 1]$. In contrast to the interval theory where an element is either part of a set or not, in fuzzy set theory the membership function $\mu$ allows us to assign a degree of membership to the elements of a set. This conceptually allows us to define a non-probabilistic approach to account for uncertainties.

The elements that are part of the set by at least a value $\alpha$ define the so-called $\alpha$-cut set

$$\boldsymbol{X}^F_\alpha = \{\boldsymbol{x} \in \boldsymbol{X} | \mu(\boldsymbol{X}) \geq \alpha\}. \tag{3}$$

If all $\alpha$-cut sets are convex, the fuzzy set is said to be convex, and each $\alpha$-cut set is exactly defined by an interval variable

$$\boldsymbol{X}^I_\alpha = \left[\boldsymbol{X}^L_\alpha, \boldsymbol{X}^U_\alpha\right]. \tag{4}$$

A fuzzy number can be thought of as exactly defined by an infinite number of $\alpha$-cuts, whereas a finite number of cuts $\boldsymbol{X}_{\alpha^k}$, $k = 1, \dots, r$ yields an approximation of $\boldsymbol{X}^F$. This concept provides information of fuzzy uncertainties by simply relying on interval theory.

For example we can obtain an approximation of the fuzzy output of an operation $\boldsymbol{Y}^F = \boldsymbol{f}(\boldsymbol{A}^F, \boldsymbol{B}^F)$ between two fuzzy numbers by using $\alpha$-level optimization [7]. Assume that each element $i = 1, \dots, d$ in $\boldsymbol{A}^F$ and $\boldsymbol{B}^F$ has the same $\alpha$-level representation $A_{i,\alpha^k}$ and $B_{i,\alpha^k}$ where $k = 1, \dots, r$. The resulting $\alpha$-cuts of the output $Y_{i,\alpha^k}$ are then exactly defined by its smallest and largest element

$$\begin{aligned} \boldsymbol{Y}_{\alpha^k} &= [\boldsymbol{Y}^L_{\alpha^k}, \boldsymbol{Y}^U_{\alpha^k}], \\ \text{with} \quad Y^L_{i,\alpha^k} &= \min_{\boldsymbol{a} \in \boldsymbol{A}_{\alpha^k}, \boldsymbol{b} \in \boldsymbol{B}_{\alpha^k}} f_i(\boldsymbol{a}, \boldsymbol{b}), \\ Y^U_{i,\alpha^k} &= \max_{\boldsymbol{a} \in \boldsymbol{A}_{\alpha^k}, \boldsymbol{b} \in \boldsymbol{B}_{\alpha^k}} f_i(\boldsymbol{a}, \boldsymbol{b}). \end{aligned} \tag{5}$$

In an equivalent way to interval fields we are also able to define space- and time-dependent fuzzy fields as the pair $(\boldsymbol{F}(\boldsymbol{x}, t), \mu)$ where $\mu : \boldsymbol{F}(\boldsymbol{x}, t) \rightarrow [0, 1]$ (for example see Fig. 1). For more information on fuzzy set theory we refer to Dubois et al. [44].

Consider a fuzzy PDE of the form

$$\tilde{\boldsymbol{G}}(\boldsymbol{x}, t, \boldsymbol{u}, \mathcal{F}) = \boldsymbol{0} \tag{6}$$

where $\boldsymbol{u} \in \mathbb{R}^h$ is the primary vector field of interest, $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^{dim}$ is the spatial field with $\Omega$ being the domain of definition of the PDE, $t \in \mathbb{R}$ is the time coordinate, $\tilde{\boldsymbol{G}}$ is an operator that maps into $\mathbb{R}^{N_G}$ and $\mathcal{F}$ is a set of convex fuzzy fields $\mathcal{F} = \{(F_1(\boldsymbol{x}, t), \mu_1) \dots, (F_s(\boldsymbol{x}, t), \mu_s)\}$. At a given $\alpha$-cut value this fuzzy PDE can be transformed into an interval PDE defined as

$$\boldsymbol{G}(\boldsymbol{x}, t, \boldsymbol{u}, \mathcal{P}_\alpha) = \boldsymbol{0} \tag{7}$$

where $\boldsymbol{G}$ still maps into $\mathbb{R}^{N_G}$ and $\mathcal{P}_\alpha$ is now the set of $\alpha$-cut interval fields found at $\mu(F_i(\boldsymbol{x}, t)) = \alpha$, $\forall i = 1, \dots, s$

$$\mathcal{P}_\alpha = \{P_1(\boldsymbol{x}, t), \dots, P_s(\boldsymbol{x}, t)\} \tag{8}$$

and each $P_i(\boldsymbol{x}, t)$ is defined by $P_i(\boldsymbol{x}, t) \in P^I_i(\boldsymbol{x}, t) = \left[P^L_i(\boldsymbol{x}, t), P^U_i(\boldsymbol{x}, t)\right]$. The solution $\tilde{\boldsymbol{u}}(\boldsymbol{x}, t, \mathcal{P}_\alpha)$ to problem (7) is complicated and can typically

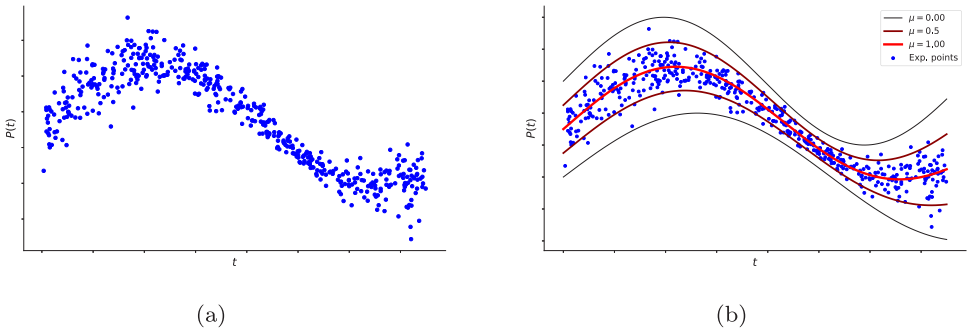



**Fig. 1.** Visual explanation of fuzzy field concept. (a) experimental points, and (b) possible resulting membership contours of approximated fuzzy field.

only be obtained approximately. Consider the time and spatial domain to be discretized into $n$ and $m$ points respectively. In this work we are interested in the two solutions of the problem that are given by $\tilde{\boldsymbol{u}}_{min}(\boldsymbol{x},t) = \tilde{\boldsymbol{u}}(\boldsymbol{x},t,\mathcal{P}_{min})$ and $\tilde{\boldsymbol{u}}_{max}(\boldsymbol{x},t) = \tilde{\boldsymbol{u}}(\boldsymbol{x},t,\mathcal{P}_{max})$ where

$$\begin{aligned}\mathcal{P}_{min} &= \min_{\mathcal{P}_\alpha^\star \in \mathcal{P}_\alpha} \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{h} \tilde{u}_k(\boldsymbol{x}_i,t_j,\mathcal{P}_\alpha^\star)\\ \mathcal{P}_{max} &= \max_{\mathcal{P}_\alpha^\star \in \mathcal{P}_\alpha} \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{h} \tilde{u}_k(\boldsymbol{x}_i,t_j,\mathcal{P}_\alpha^\star)\end{aligned} \tag{9}$$

with $\tilde{u}_k$ the $k$th component of the vector. Hence, we are interested in the two solutions which correspond to the two cases inside the allowed interval set in which the input parameter values minimize and maximize the sum of the parameters of interest with

$$\begin{aligned}\mathcal{P}_{min} &= \{P_1^{min}(\boldsymbol{x},t),\dots,P_s^{min}(\boldsymbol{x},t) | P_i^L(\boldsymbol{x},t)\\ &\quad \le P_i^{min}(\boldsymbol{x},t) \le P_i^U(\boldsymbol{x},t), \forall i = 1,\dots,s\},\\ \mathcal{P}_{max} &= \{P_1^{max}(\boldsymbol{x},t),\dots,P_s^{max}(\boldsymbol{x},t) | P_i^L(\boldsymbol{x},t)\\ &\quad \le P_i^{max}(\boldsymbol{x},t) \le P_i^U(\boldsymbol{x},t), \forall i = 1,\dots,s\}.\end{aligned} \tag{10}$$

In the limit, $\tilde{\boldsymbol{u}}_{min}(\boldsymbol{x},t)$ and $\tilde{\boldsymbol{u}}_{max}(\boldsymbol{x},t)$ will yield the output interval bounds. From an engineering point of view these two extreme cases are important because they represent the worst case scenarios of the problem at hand which has significant impact on the design of structures, for example.

## 3. Interval and fuzzy physics-informed neural networks

Neural networks are in general composed of one input, one output and $n_D - 1$ hidden layers. Let the weights and biases of the $k$th layer be denoted by $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$. Consider that the $k$th hidden layer transfers some output $\boldsymbol{x}^k$ to the $(k+1)$th layers which applies an affine transformation

$$\mathcal{L}(\boldsymbol{x}^k) = \boldsymbol{W}^{k+1}\boldsymbol{x}^k + \boldsymbol{b}^{k+1} \tag{11}$$

and some activation function $\sigma$ to it. Since, Eq. (11) is applied in every layer of the network an input $\boldsymbol{x}$ yields a network output of the form

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = (\mathcal{L}_k \circ \sigma \circ \mathcal{L}_{k-1} \circ \cdots \circ \sigma \circ \mathcal{L}_1)(\boldsymbol{x}) \tag{12}$$

where $\circ$ is a composition operator. The goal of neural networks is to find the optimal set of trainable parameters $\boldsymbol{\Theta} = \{\boldsymbol{W}^k, \boldsymbol{b}^k\}_{k=1}^{n_D}$ such that the network $\mathcal{N}(\boldsymbol{\Theta})$ provides the best fit for the input–output mapping. For more information on neural networks we refer to Goodfellow et al. [45].

The proposed interval physics-informed neural network (iPINN) consists of two separate feedforward neural networks one to approximate the maximum and minimum solution outputs termed $\mathcal{N}_u(\boldsymbol{\Lambda})$ and one to find the corresponding input fields $\mathcal{N}_P(\boldsymbol{\Psi})$ where $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ are the trainable parameters, see Fig. 2. Both networks define a global shape function over the time–space domain $(\boldsymbol{x},t)$ of the problem. We

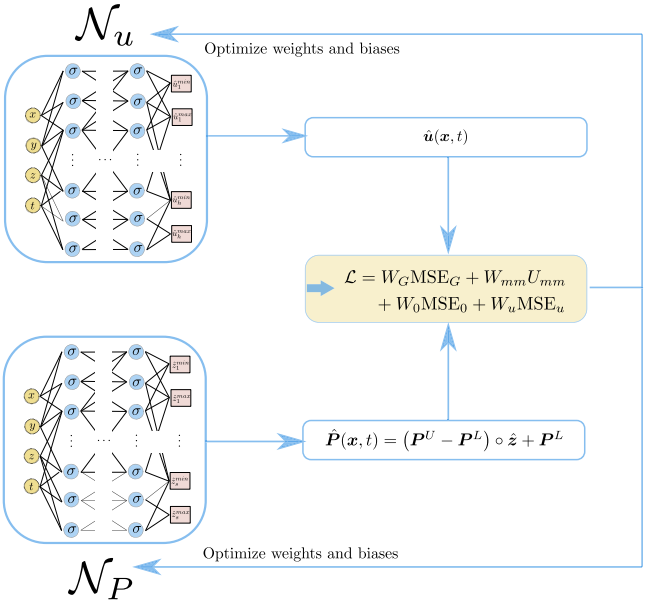



**Fig. 2.** Overview of iPINN framework.

define the final activation function of $\mathcal{N}_u(\boldsymbol{\Lambda})$ as identity whereas the equivalent function of $\mathcal{N}_P(\boldsymbol{\Psi})$ is of sigmoidal form. This is done to restrict the outputs $\hat{\boldsymbol{z}}(\boldsymbol{x},t,\boldsymbol{\Lambda})$ of $\mathcal{N}_P(\boldsymbol{\Psi})$ to be between 0 and 1 in order to allow for easy scaling of the outputs into the required interval field, i.e.

$$\hat{\boldsymbol{P}}(\boldsymbol{x},t) = \left(\boldsymbol{P}^U(\boldsymbol{x},t) - \boldsymbol{P}^L(\boldsymbol{x},t)\right)\circ\hat{\boldsymbol{z}}(\boldsymbol{x},t,\boldsymbol{\Lambda}) + \boldsymbol{P}^L(\boldsymbol{x},t), \tag{13}$$

where $\boldsymbol{P}^U(\boldsymbol{x},t) = \left[P_1^U, P_1^U, \dots, P_i^U, P_i^U, \dots, P_s^U, P_s^U\right]^T$ and $\boldsymbol{P}^L$ is defined equivalently but with the lower limit of the interval. The output of $\mathcal{N}_u(\boldsymbol{\Lambda})$ is of the form $\hat{\boldsymbol{u}}(\boldsymbol{x},t,\boldsymbol{\Lambda}) = \left[\hat{u}_1^{min}, \hat{u}_1^{max}, \dots \hat{u}_h^{min}, \hat{u}_h^{max}\right]^T$. As typically found in PINN [37] consider a set of domain training points $\{\boldsymbol{X}_{\mathcal{R}}^i\}_{i=1}^{N_{\mathcal{R}}}$, boundary training points $\{\boldsymbol{X}_u^i\}_{i=1}^{N_u}$ and initial state training points $\{\boldsymbol{X}_0^i\}_{i=1}^{N_0}$ and some time discretization $\{t^i\}_{i=1}^{N_t}$ where $N_{\mathcal{R}}$, $N_u$, $N_0$ and $N_t$ denote the number of points, respectively. We can then formulate the optimization problem for iPINN in the following way

$$\begin{aligned}(\boldsymbol{\Lambda}^\star,\boldsymbol{\Psi}^\star) = \arg\min_{(\boldsymbol{\Lambda},\boldsymbol{\Psi})} \quad & W_G\,\mathrm{MSE}_G\left(\{\{\boldsymbol{X}_{\mathcal{R}}^i\}_{i=1}^{N_{\mathcal{R}}},\{t^j\}_{j=1}^{N_t}\},(\boldsymbol{\Lambda},\boldsymbol{\Psi})\right)\\ &+ W_{mm}\,U_{mm}\left(\{\{\boldsymbol{X}_{\mathcal{R}}^i\}_{i=1}^{N_{\mathcal{R}}},\{t^j\}_{j=1}^{N_t}\},\boldsymbol{\Lambda}\right)\\ &+ W_0\,\mathrm{MSE}_0\left(\{\{\boldsymbol{X}_t^i\}_{i=1}^{N_t},t^0\},\boldsymbol{\Lambda}\right) +\\ &W_u\,\mathrm{MSE}_u\left(\{\{\boldsymbol{X}_u^i\}_{i=1}^{N_u},\{t^j\}_{j=1}^{N_t}\},\boldsymbol{\Lambda}\right)\end{aligned} \tag{14}$$

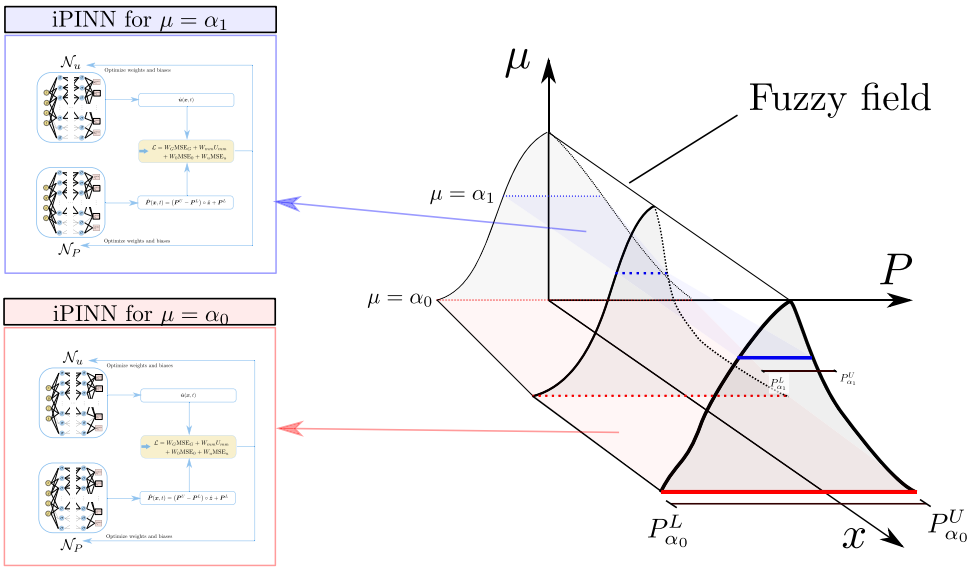



**Fig. 3.** Overview of fPINN framework as a collection of iPINNs.

where

$$\begin{aligned}
\mathrm{MSE}_G\left(\{\{\boldsymbol{X}_{\mathcal{R}}^i\}_{i=1}^{N_{\mathcal{R}}},\{t^j\}_{j=1}^{N_t}\},(\boldsymbol{\Lambda},\boldsymbol{\Psi})\right) &= \frac{1}{N_{\mathcal{R}}N_t}\sum_{i=1}^{N_{\mathcal{R}}}\sum_{j=1}^{N_t}\sum_{k=1}^{N_G} \\
&\times \left|G_k(\boldsymbol{X}_{\mathcal{R}}^i,t^j,\hat{\boldsymbol{u}}(\boldsymbol{\Lambda}),\hat{\boldsymbol{P}}(\boldsymbol{\Psi}))\right|^2 \\
U_{mm}\left(\{\{\boldsymbol{X}_{\mathcal{R}}^i\}_{i=1}^{N_{\mathcal{R}}},\{t^j\}_{j=1}^{N_t}\},\boldsymbol{\Lambda}\right) &= \sum_{i=1}^{N_{\mathcal{R}}}\sum_{j=1}^{N_t}\sum_{k=1}^{h}\hat{u}_k^{min}(\boldsymbol{X}_R^i,t^j) \\
&- \sum_{i=1}^{N_{\mathcal{R}}}\sum_{j=1}^{N_t}\sum_{k=1}^{h}\hat{u}_k^{max}(\boldsymbol{X}_R^i,t^j) \\
\mathrm{MSE}_0\left(\{\{\boldsymbol{X}_0^i\}_{i=1}^{N_0},t^0\},\boldsymbol{\Lambda}\right) &= \frac{1}{N_0}\sum_{i=1}^{N_0}\sum_{k=1}^{h} \\
&\times \left|\hat{u}_k^{min}(\boldsymbol{X}_0^i,t^0)-u_k(\boldsymbol{X}_0^i,t^0)\right|^2 \\
&+ \left|\hat{u}_k^{max}(\boldsymbol{X}_0^i,t^0)-u_k(\boldsymbol{X}_0^i,t^0)\right|^2 \\
\mathrm{MSE}_u\left(\{\{\boldsymbol{X}_u^i\}_{i=1}^{N_u},\{t^j\}_{j=1}^{N_t}\},\boldsymbol{\Lambda}\right) &= \frac{1}{N_uN_t}\sum_{i=1}^{N_u}\sum_{j=1}^{N_t}\sum_{k=1}^{h} \\
&\times \left|\hat{u}_k^{min}(\boldsymbol{X}_u^i,t^j)-u_k(\boldsymbol{X}_u^i,t^j)\right|^2 \\
&+ \left|\hat{u}_k^{max}(\boldsymbol{X}_u^i,t^j)-u_k(\boldsymbol{X}_u^i,t^j)\right|^2
\end{aligned} \tag{15}$$

and $W_G$, $W_{mm}$, $W_0$ and $W_u$ are user-chosen weights.

To clarify the intuition behind this particular choice for the loss function, an explanation of its constituents is provided next. $\mathrm{MSE}_G$ is the term that needs to be minimized in order to ensure that the solution $\boldsymbol{u}$ accurately satisfies the PDE at the interior of its domain $\Omega$. In a similar fashion, the purpose of $\mathrm{MSE}_0$ and $\mathrm{MSE}_u$ is to impose the initial and boundary conditions, respectively, on the solution. Lastly, $U_{mm}$ is the additional term in the iPINN framework, whose minimization eventually yields the extrema $\hat{\boldsymbol{u}}^{min}$ and $\hat{\boldsymbol{u}}^{max}$.

A fuzzy physics-informed neural network (fPINN) can then be understood as a collection of iPINNs, see Fig. 3.

After minimization of the loss function of Eq. (15), the trainable parameters of $\mathcal{N}_u$ and $\mathcal{N}_P$ are obtained. The outputs of these networks yield approximate bounds to iPDE solutions ($\mathcal{N}_u$) as well as the corresponding input fields ($\mathcal{N}_P$). It needs to be highlighted that in contrast to FEM no correlation length specification of the input fields needs to be assumed and Monte-Carlo-like simulations are also not required. Additionally, we retain all major advantages of the PINN approach, i.e. the problem is solved meshfree and inverse problems

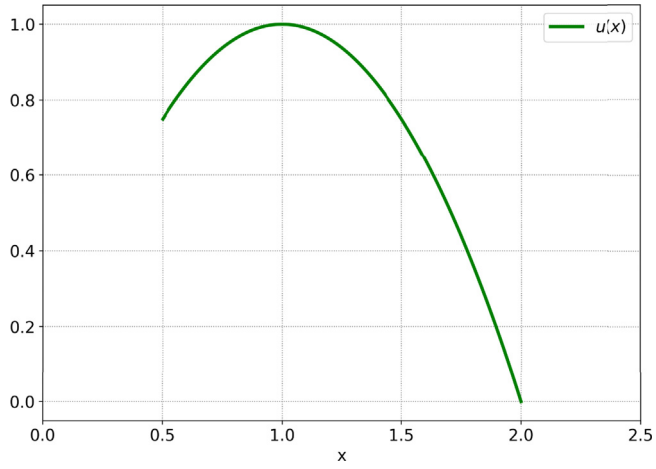
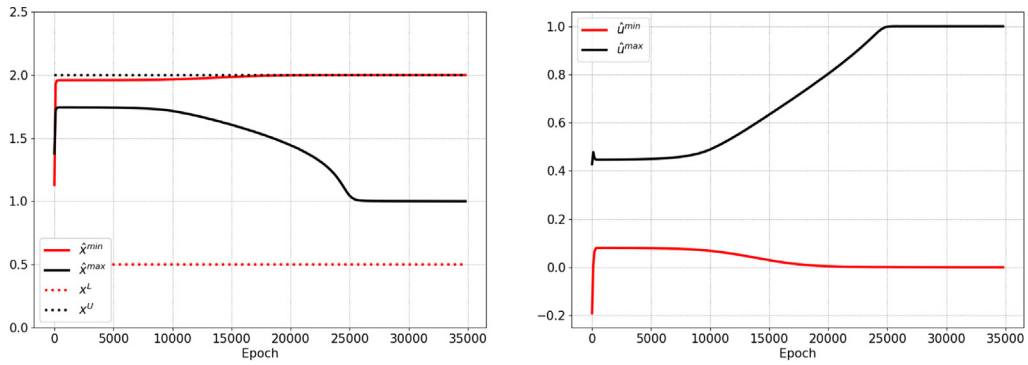



**Fig. 4.** Dependence of primary variable over elements of interval variable.

are straightforwardly set up and solvable. For more information on PINNs we refer to Raissi et al. [37]. One slightly negative aspect of this approach is that we employ two network outputs for $\mathcal{N}_u$ and $\mathcal{N}_P$ for each input field which for a larger number of input fields might be restrictive. However this limitation is dependent on the problem complexity and its dimensionality. The presented approach is somewhat related to the stochastic PINN framework proposed in [42]. However their main concern is probabilistic uncertainty quantification which is not the focus of this paper.

## 4. Application

In the following, the proposed formulation is tested and studied on a set of problems of increasing complexity. First, a simple educational problem is considered, involving an interval parameter and fuzzy parameter, and thereafter, we attempt to find the solution to a more complex one-dimensional structural problem involving two simultaneously applied interval fields. Lastly, we study the performance of the framework on a time-dependent interval PDE problem. The fPINN/iPINN formulation was implemented in Pytorch [46] and the




(a) Predicted and constraining input field (b) Predicted primary output bounds

**Fig. 5.** Results of introductory example. Outputs of both trained networks over training iterations.

network parameters were optimized using the Adam optimizer [47].[1] The FEM results for the second problem were obtained using the FEniCS framework [48].

### 4.1. Introductory problem: Function of an interval and fuzzy parameter

Consider the following simple functional problem with an interval parameter

$$u^I(x^I) = x^I(2 - x^I), \tag{16}$$

where the interval parameter is constrained by

$$x^I = [0.5, 2.0]\,. \tag{17}$$

The primary output variable $u^I(x)$ is plotted over the elements of the interval input in Fig. 4.

It can be seen that $u$ is non-monotonically dependent on $x$. As a result the output bounds $u^{min}$ and $u^{max}$ are not just resulting from endpoint combinations, i.e. just the evaluation of $x = x^L = 0.5$ and $x = x^U = 2.0$ is not enough to obtain the output bounds for more information see e.g. [13]. In fact as seen in Fig. 4 the bounds of the primary variable $u^{min} = 0.0$ and $u^{max} = 1.0$ are results of the input interval values $x^{min} = 2.0$ and $x^{max} = 1.0$ respectively.

To highlight how iPINN is able to solve this problem we consider both $\mathcal{N}_u$ and $\mathcal{N}_P$ consisting of 2 hidden layers with 20 neurons. The activation functions are chosen to be the hyperbolic tangent function. These choices are arbitrary and are not the result of any hyperparameter optimization, i.e. they do not represent any special network setup to the best of the author's knowledge. Both networks have 1 input and 2 outputs. When dealing with interval fields and iPDEs the spatial and temporal values are the inputs. Since in this simple problem the primary variable is independent of space and time, a constant arbitrary value can be chosen as the input which has no effect on the solution (we chose 1.0). The training is conducted with a learning rate of $1e{-}3$ and is stopped after 35,000 iterations. High emphasis is put on the fulfillment of the residual $u^I(x^I) - x^I(2 - x^I) = 0$. Hence the loss weights in Eq. (14) are chosen to be $W_G = 100,000$ and $W_{mm} = 1$. Since there are no initial and boundary conditions we do not need to account for those terms in the loss function.

Fig. 5a shows the constraining limit values of the input interval $x^L$ and $x^U$ (compare with Eq. (17)) highlighted by the dashed lines, over the training process. Additionally, both outputs of $\mathcal{N}_P$ denoted $\hat{x}^{min}$ and $\hat{x}^{max}$ are also plotted in Fig. 5a. These describe the two values the input $x^I$ has to take such that $u^I(x^I = \hat{x}^{max})$ and $u^I(x^I = \hat{x}^{min})$ result in the possible maximum and minimum values of $u^I$, respectively. These two extreme cases of $u^I$ are approximated by the output of the second network $\mathcal{N}_u$ denoted by $\hat{u}^{max}$ and $\hat{u}^{min}$ which are plotted over the training process in Fig. 5b.

Overall, it can be seen that both the input values, as well as the output bounds are accurately predicted after around 25,000 epochs, i.e. $\hat{x}^{max} \to 1.0$, $\hat{x}^{min} \to 2.0$, $\hat{u}^{min} \to 0.0$ and $\hat{u}^{max} \to 1.0$.

Instead of an interval number, consider the function to be dependent on a fuzzy number

$$u^F(x^F) = x^F(2 - x^F). \tag{18}$$

Let $x^F$ be a triangular fuzzy number of the form given in Fig. 6 where $\mu(x = 1) = 1.0$. We can see that the previously investigated problem coincides with the $\alpha$-cut at $\mu = 0$. In order to obtain an approximate description of the output fuzzy bounds we consider 4 more $\alpha$-cuts at $0.25, 0.5, 0.75$ and 1.0. The interval bounds for these cuts are also shown in Fig. 6.

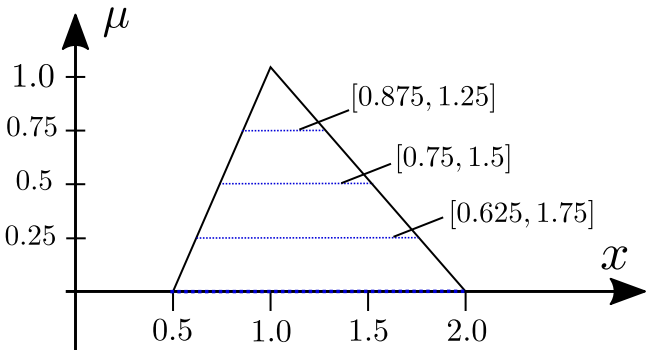
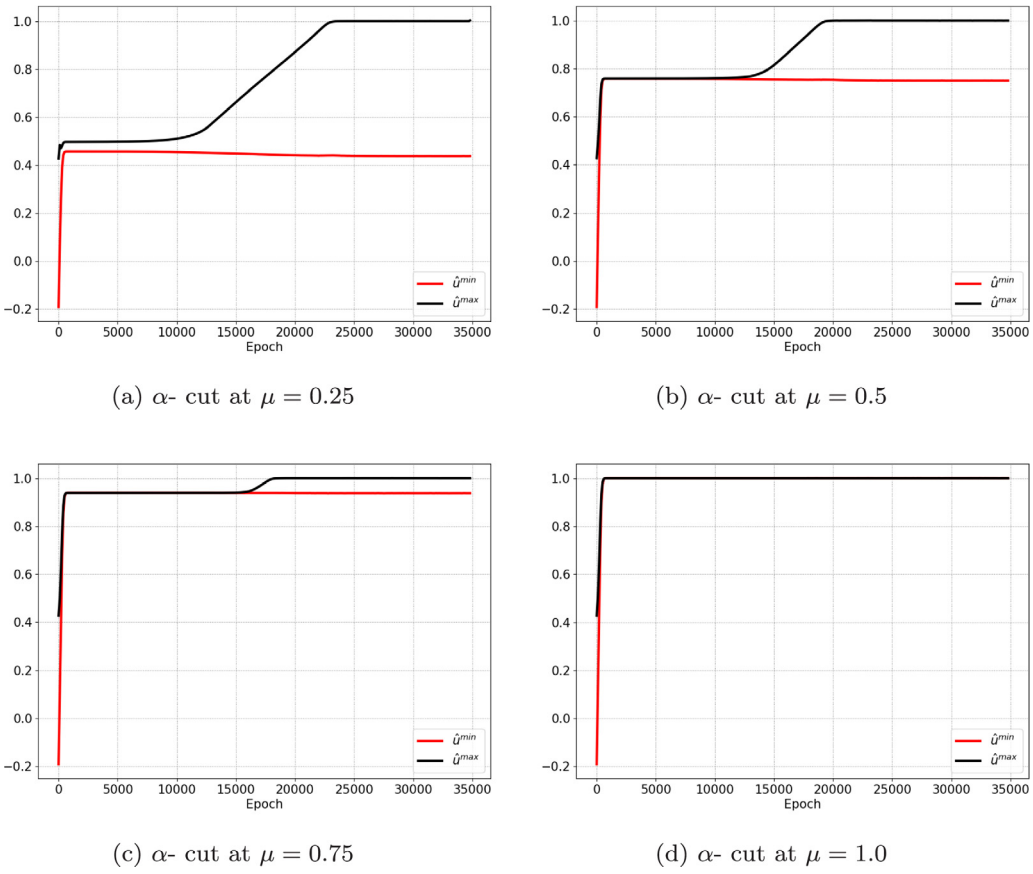



**Fig. 6.** Triangular fuzzy input with interval values at $\alpha$-cuts.

Following the fPINN concept, we build an iPINN for each new $\alpha$-cut value, i.e. resulting in 4 new iPINNs. Only changing the considered bounding interval in comparison to the previous problem and leaving the rest of the networks unchanged (hyperparameter, learning rate, etc.) the two predicted extreme values $\hat{u}^{max}$ and $\hat{u}^{min}$ which are the two outputs of $\mathcal{N}_u$ are plotted in Fig. 7 for each subproblem over the number of epochs. It can be seen that the four networks (one for each subproblem) converge to specific extreme values over the course of the training process. Naturally, for the $\alpha$-cut at $\mu = 1$ (see Fig. 7d) the convergence is reached relatively quickly since the input set $x^I$ of possible values collapses to a single element $x^I = [1, 1]$ (see Fig. 6).

Hence, using these results for the considered $\alpha$-cuts we can make an approximation of the output fuzzy number by knowing the intervals at each $\alpha$-cut. For example, knowing that $\hat{u}^{max} = 1.0$ and $\hat{u}^{min} = 0.75$ at $\mu = 0.5$ as resulting from Fig. 7b we know that the $\alpha$-cut of $u^F$ at $\mu = 0.5$ is (approximately) described by the interval $\hat{u}_{\alpha=0.5} = [0.75, 1.0]$.

[1] Codes will be made public under https://github.com/FuhgJan/intervalAndFuzzyPINN after acceptance of this paper.




(a) $\alpha$- cut at $\mu = 0.25$ (b) $\alpha$- cut at $\mu = 0.5$

(c) $\alpha$- cut at $\mu = 0.75$ (d) $\alpha$- cut at $\mu = 1.0$

**Fig. 7.** fPINN predicted bounding outputs for different $\alpha$-cuts over training process.

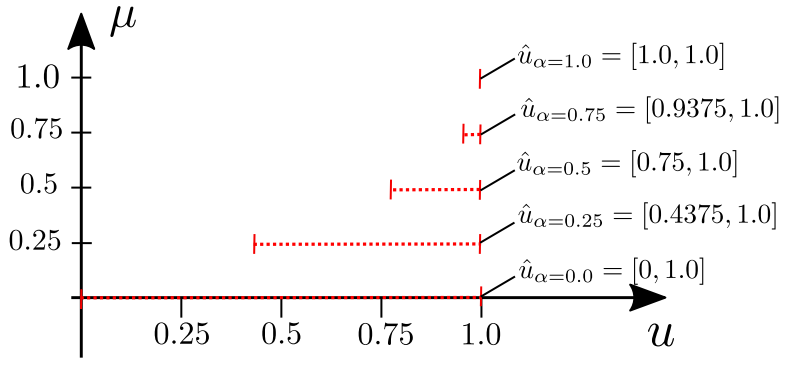



**Fig. 8.** Fuzzy (interval) output values for considered $\alpha$-cuts obtained using fPINN framework.

Therefore, using fPINN we are able to obtain the fuzzy number estimations at each of the $\alpha$-cuts as shown in Fig. 8 which exactly correspond to the analytical solutions that can be obtained trivially. An interesting side note is that the output fuzzy number seen in Fig. 8 appears to be no longer triangular.

### 4.2. One-dimensional bar with two interval fields

In order to test the capability of the proposed approach when dealing with a more realistic structural problem with interval fields, we study the iPINN on a 1D bar problem, see Fig. 9. Here, a one-dimensional bar is clamped on the left-hand side and subjected to a body force $n$ and an applied load $P$. The relevant interval ordinary

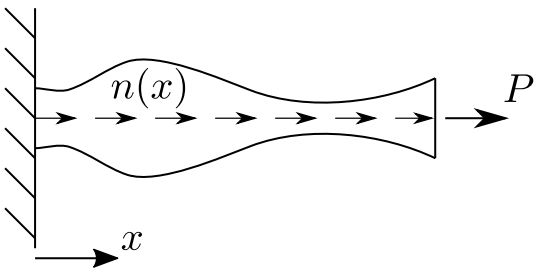



**Fig. 9.** 1D bar with area and Young's modulus as interval fields.

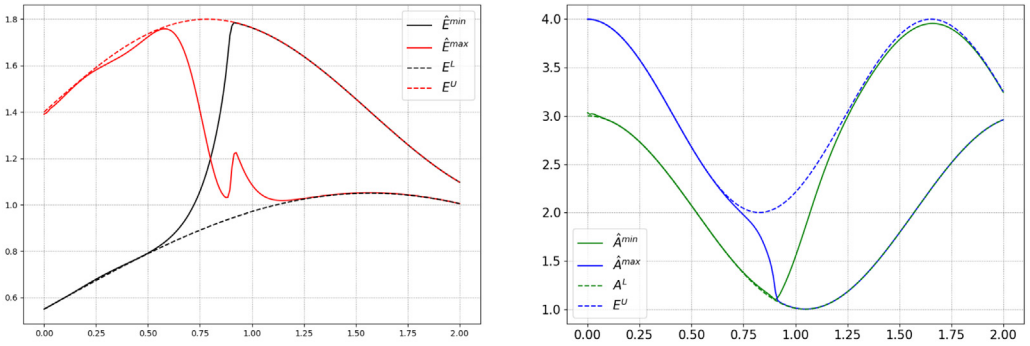



**Fig. 10.** Predicted and constraining input fields, (a) Young's modulus field and (b) area field.

differential equation is given by

$$\frac{\partial}{\partial x}\left(E^I(x)A^I(x)\frac{\partial u}{\partial x}\right)+n(x)=0 \tag{19}$$

where $n(x) = \cos(3x)x$ in N/m, $E^I$ in N/m$^2$ and $A^I$ in m$^2$ are the spatially dependent interval fields of the Young's modulus and the area which are defined as

$$E^I(x)\begin{cases}\geq E^L(x) & =0.5\sin(x)+0.55\\ \leq E^U(x) & =0.4\sin(2x)+1.4\end{cases} \tag{20}$$

and

$$A^I(x)\begin{cases}\geq A^L(x) & =\cos(3x)+2.0\\ \leq A^U(x) & =\cos(3.8x)+3.0.\end{cases} \tag{21}$$

The boundary conditions are given as

$$u(x=0)=0,\qquad \left.\frac{\partial u}{\partial x}\right|_{x=L}=\frac{P}{E(x=L)A(x=L)} \tag{22}$$

where $P = 0.1N$ and where the length of the bar is $L = 2m$. The training is conducted with a learning rate of $1e{-}4$. The network of the primary solution network $\mathcal{N}_u$ consists of 4 hidden layers with 40 neurons each while the input interval network $\mathcal{N}_P$ has 5 hidden layers with 50 neurons. The authors cannot find anything special about these hyperparameter values. They are employed without any immediate considerations. We use 200 equidistant points inside the one-dimensional training domain. The FEM mesh consists of 200 elements. During the studies of this work it was found that high-emphasis needs to be placed on the fulfillment of the residual, i.e. $W_G = 100,000$ in Eq. (14) whereas the other weights are less influential for this study $W_{mm} = W_0 = W_u = 1$. The network is trained for 500,000 epochs.

Due to the unique formulation of iPINN we are able to obtain the input fields leading to the bounds of the interval displacement field as a simple byproduct of the training approach, see Fig. 10. Fig. 10a shows the upper and lower bounds of the Young's modulus as given in Eq. (20) (dashed lines) and the input fields (solid lines) that lead to the extreme values of $u$. The equivalent fields of interest are plotted in Fig. 10b for the area $A$.

It can be seen that the predicted input fields are effectively equal to the constraining interval values for the main part of the computational domain. However both input field predictions "switch" from being closely aligned to the lower bound to being closely aligned to the upper bound value and vice versa around the center of the bar. This is an interesting observation which is not trivially predictable beforehand.

The bounds of the interval displacement field resulting from the fully trained outputs of $\mathcal{N}_P$ are shown in Fig. 11a represented by the two solid lines. Using FEM, the output bounds of the variable of interest obtained by combining one-by-one the four possible combinations of input interval limits ($E^L - A^L, E^U - A^L, E^L - A^U, E^U - A^U$) are highlighted as dashed lines in the figure. The proposed framework provides solution bounds of the primary variable that are more proficient than using the naive approach. Furthermore, we can see that the "switch" of the input fields occurs around the global minimum of the primary variable.

The absolute error between the FEM solution and the iPINN output for the input fields obtained through $\mathcal{N}_P$ (i.e. $\hat{E}^{max}$, $\hat{E}^{min}$, $\hat{A}^{max}$ and $\hat{A}^{min}$ from Fig. 10) are shown in Fig. 11b which proves that the approach is able to effectively solve the problem in a similar accuracy to FEM.

The total loss and all individual losses over the training process are shown in Fig. 12. It can be seen that the losses decline in a satisfactory manner and that especially the residual losses ($\text{MSE}_G$ and the boundary loss) are fulfilled quite accurately due to the choice of the loss weights as described above. In contrast to FEM approaches we do not need to sample the input fields, i.e. we do not need to assume any spatial correlations of the interval fields. Furthermore, we also do not need to conduct Monte-Carlo sampling as a means to obtain approximations for the output bounds.

### 4.3. Interval nonlinear partial differential equation

Finally, we investigate the performance of the iPINN framework on the following time-dependent PDE

$$\frac{\partial u}{\partial t}=0.01u\frac{\partial^2 u}{\partial x^2}-k^I(x,t)u^3+\left(k^I(x,t)\right)^3 \tag{23}$$

with the boundary conditions

$$\begin{aligned}&u(-1,t)=u(1,t)=0\\ &u(x,0)=1-x^2.\end{aligned} \tag{24}$$

Here, $k^I(x,t)$ represents a temporally and spatially varying interval field. Assume the spatial components to be defined by $x \in [-1,1]$ and the time to be restricted to $t \in [0,1]$.

Suppose we are able to measure $k(x,t)$ the temporal and spatial variation of the field of interest. These experimentally obtained points are shown in Fig. 13a. Furthermore, assume that from these points we are able to define lower and upper bounding limits of $k(x,t)$ (Fig. 13b) given by

$$k^I(x,t)\begin{cases}\geq k^L(x,t) & =0.5\sin(3x)\cos(t),\\ \leq k^U(x,t) & =\sin(3x)\cos^2(t)+3.\end{cases} \tag{25}$$

Both, the input interval network $\mathcal{N}_P$ and the primary solution network $\mathcal{N}_u$ are chosen to have 3 hidden layers with 40 neurons. Here, we iterate again that we cannot find anything special about these hyperparameter values. They are employed without any immediate

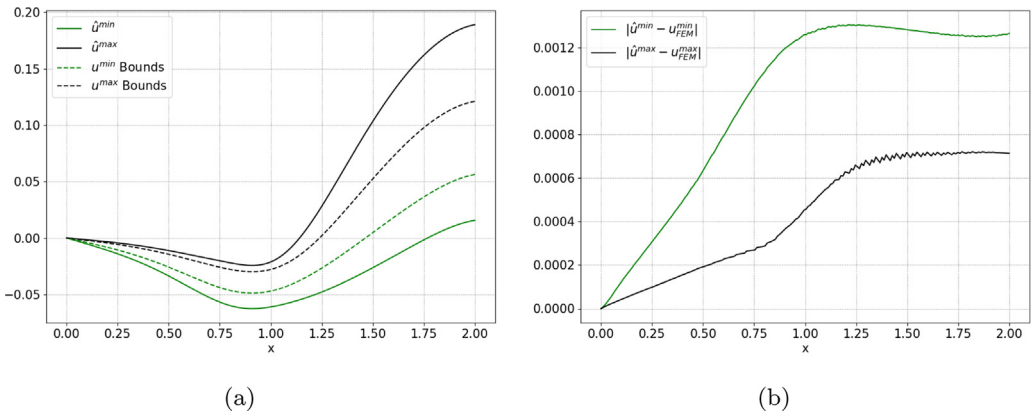



**Fig. 11.** Comparisons and output of primary output. (a) $\mathcal{N}_u$ output and maximum and minimum FEM outputs using 4 combination of interval limits, and (b) Error between FEM and the $\mathcal{N}_u$ output for $\mathcal{N}_P$ input field.

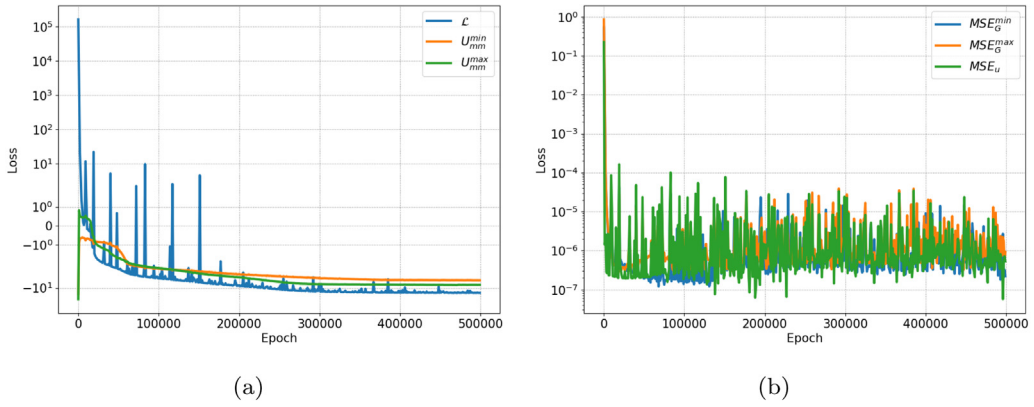



**Fig. 12.** Losses over the training process. (a) Non-residual losses, and (b) Residual losses.

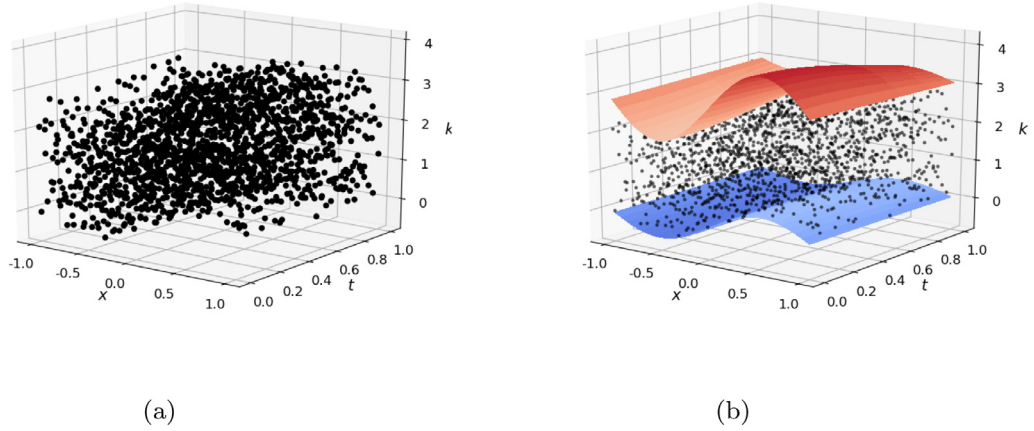
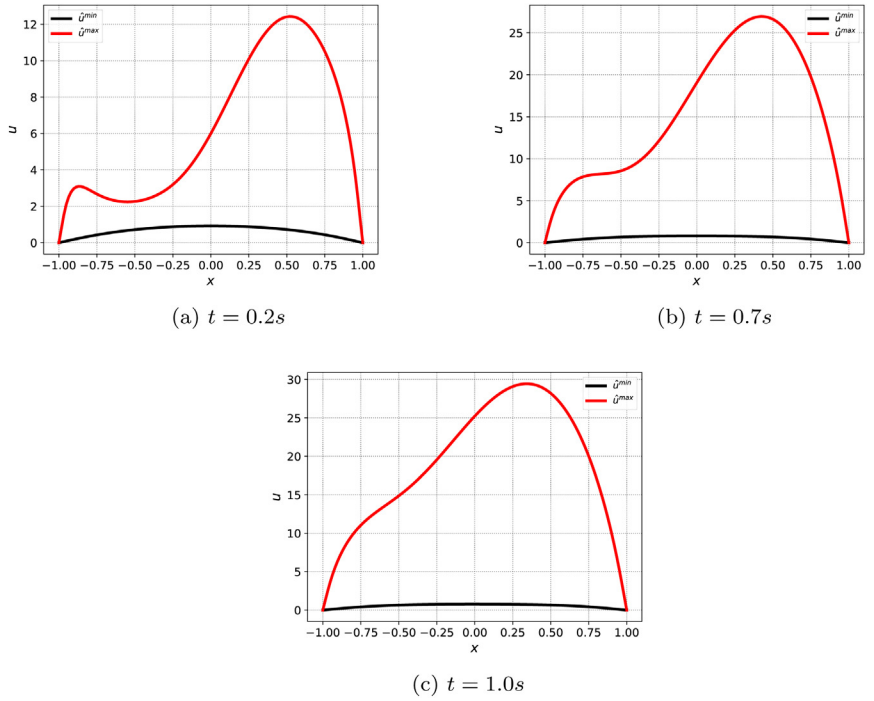



**Fig. 13.** Nonlinear interval PDE problem. (a) "Experimentally" obtained points and (b) fitted lower and upper limits.

considerations. A learning rate of $1e{-}4$ for the ADAM optimizer is employed. We discretize the time domain with 50 and the spatial domain with 125 equidistant points. Finally, we choose $W_G = W_0 = 100,000$ and all other weights equal to 1.

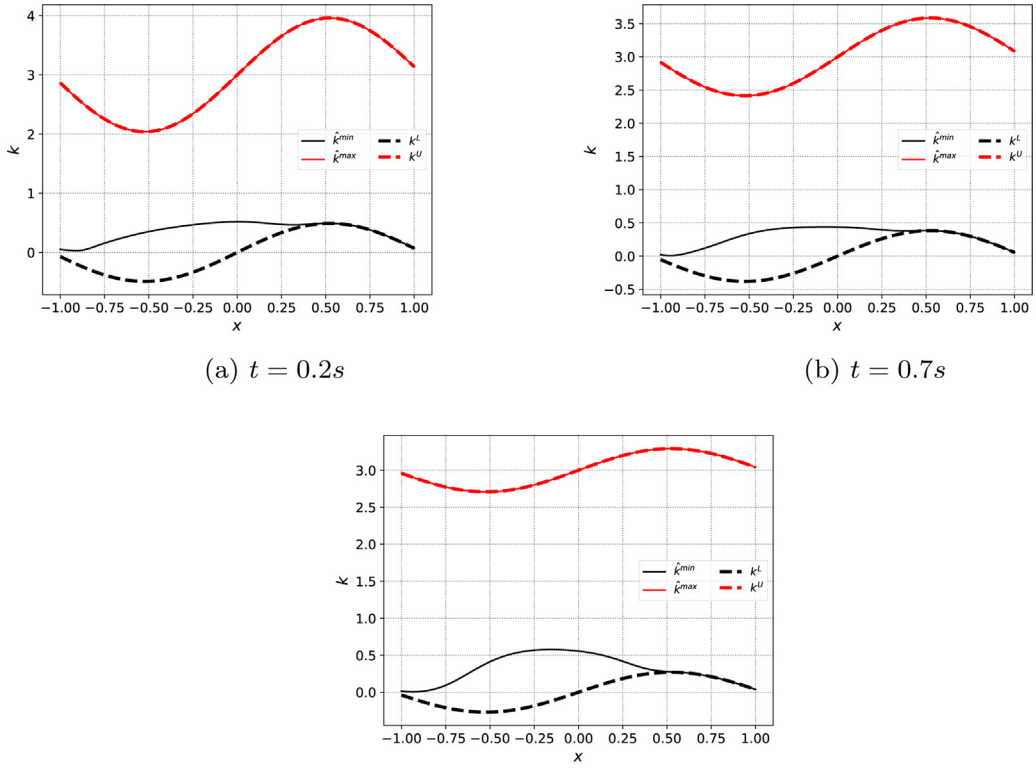
The maximum and minimum bounding primary output fields are plotted for three different points in time in Fig. 14. It can be seen that due to the interval parameter $k^I$ in Eq. (23), the primary output can take drastically different profiles. Upon solution, the corresponding bounding fields of the inputs for these three time points accompanied by the allowed upper and lower limits $k^U$ and $k^L$ are shown in Fig. 15. It can be noticed that $\hat{k}^{max}$, the approximated maximum field value, approaches the upper limit $k^U$. On the other hand, the approximated minimum field value $\hat{k}^{min}$ does not exactly correspond to the lower bound. In order to find this field an appropriate sampling technique would have been required if traditional approaches such as Monte-Carlo would have been used to solve the problem. With iPINN this field is directly available to us after solving the optimization problem of Eq. (14). Fig. 16 compares the primary output field solutions of Eq. (23) when $k^I = \hat{k}^{min}$ and $k^I = \hat{k}^L$ to highlight that the lower bounding limit does indeed not yield the minimum primary output field. Lastly, the non-residual and the residual errors of the problem over the training process are shown in Fig. 17. It can be seen that after around 100,000 epochs the non-residual values have converged to a stable value (Fig. 17a) where $U^{min}_{mm}$ and $U^{max}_{mm}$ correspond to the




(a) $t = 0.2s$ (b) $t = 0.7s$

(c) $t = 1.0s$

**Fig. 14.** Maximum and minimum predicted output fields for the nonlinear PDE problem for different points in time. Note the change in axis values.




(a) $t = 0.2s$ (b) $t = 0.7s$

(c) $t = 1.0s$

**Fig. 15.** Predicted and constraining input fields for the nonlinear PDE problem for specific time instances. Note the change in axis values.

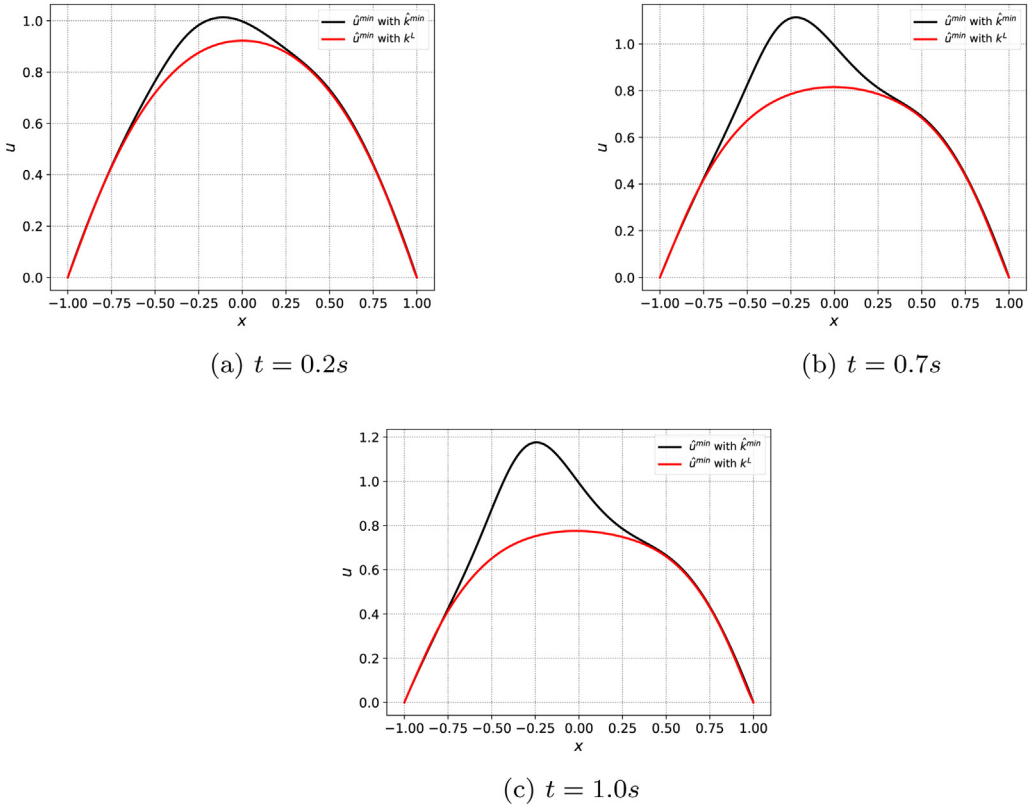



**Fig. 16.** Minimum predicted output field with corresponding minimum input field vs. predicted output field using the minimum bound $k^L$ for nonlinear PDE problem.

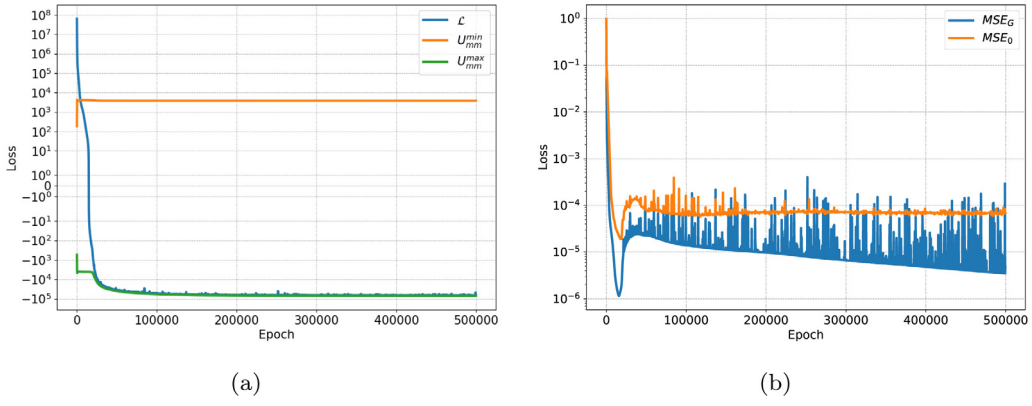



**Fig. 17.** (a) Non-residual losses over training epochs and (b) residual losses over training epochs for nonlinear interval PDE.

minimum and maximum field components of $U_{mm}$. Furthermore, after 100,000 epochs the initial $MSE_0$ and PDE residuals $MSE_G$ have both reached very accurate values lower than $1e{-}4$, see Fig. 17b.

## 5. Discussion and outlook

The presented paper proposes a physics-informed neural network formulation for solving interval PDEs which can be used to obtain approximate solutions to fuzzy PDEs with convex fuzzy sets. The proposed network infrastructure, termed interval physics-informed neural networks (iPINNs), consists of two separate feedforward neural networks with one aiming to approximate minimum and maximum possible primary outputs, and the other one aiming to obtain the accompanying input fields that lead to the specific solutions of the PDE. The presented approach is studied in an introductory example as well as in a one-dimensional structural problem and in a nonlinear time-dependent PDE, showing promising results. In particular the input fields that correspond to the bounds of the interval displacement field are obtained as a byproduct of the iPINN formulation without any necessary prior knowledge of the spatial correlation of the input field. We believe this framework has a lot of potential by being able to circumvent some of the problems of the finite element method when dealing with interval and fuzzy fields.

It should be highlighted again that the solutions of fuzzy and interval PDEs provide non-probabilistic output fields. These approaches are useful when a cumulative distribution function cannot be reliably identified, which might be the case when physical or numerical experiments are either too costly or too time-consuming.

On the other hand, using usual stochastic frameworks the PDE solutions are probabilistic, which require precise probabilistic characteristics, including the probability distribution function and the correct covariance function. When these are known, probabilistic approaches offer a powerful and convenient methodology to account for uncertainties.

In the future we aim to extend this framework to higher dimensions and try to tackle inverse problems related to finding possible input intervals from observed output ranges. Furthermore we could potentially study more theoretical problems that are hard to study with interval finite element methods, i.e. physical law discovery and higher dimensional nonlinear PDEs.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] M. Ostoja-Starzewski, Random field models of heterogeneous materials, Int. J. Solids Struct. 35 (19) (1998) 2429–2455.

[2] M. Ostoja-Starzewski, Material spatial randomness: From statistical to representative volume element, Probab. Eng. Mech. 21 (2) (2006) 112–132.

[3] A. Der Kiureghian, J.-B. Ke, The stochastic finite element method in structural reliability, Probab. Eng. Mech. 3 (2) (1988) 83–91.

[4] G. Stefanou, The stochastic finite element method: past, present and future, Comput. Methods Appl. Mech. Engrg. 198 (9–12) (2009) 1031–1051.

[5] J.N. Fuhg, Adaptive surrogate models for parametric studies, 2019, arXiv preprint arXiv:1905.05345.

[6] J.N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, Arch. Comput. Methods Eng. (2020) 1–59.

[7] B. Möller, W. Graf, M. Beer, Fuzzy structural analysis using $\alpha$-level optimization, Comput. Mech. 26 (6) (2000) 547–565.

[8] H.-H. Bothe, Fuzzy Logic: Einführung in Theorie und Anwendungen, Springer-Verlag, 2013.

[9] B. Möller, W. Graf, M. Beer, Safety assessment of structures in view of fuzzy randomness, Comput. Struct. 81 (15) (2003) 1567–1582.

[10] R.E. Moore, Interval Analysis, Vol. 4, Prentice-Hall Englewood Cliffs, 1966.

[11] L. Jaulin, M. Kieffer, O. Didrit, E. Walter, Interval analysis, in: Applied Interval Analysis, Springer, 2001, pp. 11–43.

[12] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, SIAM, 2009.

[13] L. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353, http://dx.doi.org/10.1016/S0019-9958(65)90241-X, URL: https://www.sciencedirect.com/science/article/pii/S001999586590241X.

[14] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Vol. 4, Prentice hall New Jersey, 1995.

[15] H.-J. Zimmermann, Fuzzy Set Theory—and its Applications, Springer Science & Business Media, 2011.

[16] S.-H. Chen, X.-W. Yang, Interval finite element method for beam structures, Finite Elem. Anal. Des. 34 (1) (2000) 75–88.

[17] A. Sofi, E. Romeo, A novel interval finite element method based on the improved interval analysis, Comput. Methods Appl. Mech. Engrg. 311 (2016) 671–697.

[18] B. Ni, C. Jiang, Interval field model and interval finite element analysis, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112713.

[19] S.S. Rao, J.P. Sawyer, Fuzzy finite element approach for analysis of imprecisely defined systems, AIAA J. 33 (12) (1995) 2364–2370.

[20] R.L. Muhanna, R.L. Mullen, Formulation of fuzzy finite-element methods for solid mechanics problems, Comput.-Aided Civ. Infrastruct. Eng. 14 (2) (1999) 107–117.

[21] H. Yin, D. Yu, S. Yin, B. Xia, Fuzzy interval finite element/statistical energy analysis for mid-frequency analysis of built-up systems with mixed fuzzy and interval parameters, J. Sound Vib. 380 (2016) 192–212.

[22] F.N. Schietzold, A. Schmidt, M.M. Dannert, A. Fau, R.M.N. Fleury, W. Graf, M. Kaliske, C. Könke, T. Lahmer, U. Nackenhorst, Development of fuzzy probability based random fields for the numerical structural design, GAMM-Mitt. 42 (1) (2019) e201900004.

[23] D. Moens, M. De Munck, W. Desmet, D. Vandepitte, Numerical dynamic analysis of uncertain mechanical structures based on interval fields, in: IUTAM Symposium on the Vibration Analysis of Structures with Uncertainties, Springer, 2011, pp. 71–83.

[24] W. Verhaeghe, W. Desmet, D. Vandepitte, D. Moens, Interval fields to represent uncertainty on the output side of a static FE analysis, Comput. Methods Appl. Mech. Engrg. 260 (2013) 50–62.

[25] G. Muscolino, A. Sofi, M. Zingales, One-dimensional heterogeneous solids with uncertain elastic modulus in presence of long-range interactions: Interval versus stochastic analysis, Comput. Struct. 122 (2013) 217–229.

[26] A. Sofi, G. Muscolino, Static analysis of Euler–Bernoulli beams with interval Young's modulus, Comput. Struct. 156 (2015) 72–82.

[27] M.M. Dannert, M.G. Faes, R.M. Fleury, A. Fau, U. Nackenhorst, D. Moens, Imprecise random field analysis for non-linear concrete damage analysis, Mech. Syst. Signal Process. 150 (2021) 107343.

[28] A. Sofi, E. Romeo, O. Barrera, A. Cocks, An interval finite element method for the analysis of structures with spatially varying uncertainties, Adv. Eng. Softw. 128 (2019) 1–19, http://dx.doi.org/10.1016/j.advengsoft.2018.11.001, URL: https://www.sciencedirect.com/science/article/pii/S0965997818305192.

[29] M.M. Dannert, A. Fau, R.M. Fleury, M. Broggi, U. Nackenhorst, M. Beer, A probability-box approach on uncertain correlation lengths by stochastic finite element method, PAMM 18 (1) (2018) e201800114.

[30] D. Huang, J.N. Fuhg, C. Weißenfels, P. Wriggers, A machine learning based plasticity model using proper orthogonal decomposition, Comput. Methods Appl. Mech. Engrg. 365 (2020) 113008.

[31] J.N. Fuhg, C. Böhm, N. Bouklas, A. Fau, P. Wriggers, M. Marino, Model-data-driven constitutive responses: Application to a multiscale computational framework, Internat. J. Engrg. Sci. 167 (2021) 103522, http://dx.doi.org/10.1016/j.ijengsci.2021.103522, URL: https://www.sciencedirect.com/science/article/pii/S0020722521000690.

[32] J.N. Fuhg, M. Marino, N. Bouklas, Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114217.

[33] J.N. Fuhg, N. Bouklas, On physics-informed data-driven isotropic and anisotropic constitutive models through probabilistic machine learning and space-filling sampling, 2021, arXiv preprint arXiv:2109.11028.

[34] T. Kadeethum, F. Ballarin, N. Bouklas, Non-intrusive reduced order modeling of poroelasticity of heterogeneous media based on a discontinuous Galerkin approximation, 2021, arXiv preprint arXiv:2101.11810.

[35] Q. Hernandez, A. Badías, D. González, F. Chinesta, E. Cueto, Deep learning of thermodynamics-aware reduced-order models from data, Comput. Methods Appl. Mech. Engrg. 379 (2021) 113763.

[36] T. Kadeethum, D. O'Malley, J.N. Fuhg, Y. Choi, J. Lee, H.S. Viswanathan, N. Bouklas, A framework for data-driven solution and parameter estimation of PDEs using conditional generative adversarial networks, Nat. Comput. Sci. 1 (2021) 819–829, http://dx.doi.org/10.1038/s43588-021-00171-3.

[37] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[38] A.D. Jagtap, K. Kawaguchi, G.E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, J. Comput. Phys. 404 (2020) 109136.

[39] S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, J. Comput. Phys. (2021) 110768.

[40] J.N. Fuhg, N. Bouklas, The mixed deep energy method for resolving concentration features in finite strain hyperelasticity, J. Comput. Phys. 451 (2022) 110839, http://dx.doi.org/10.1016/j.jcp.2021.110839, URL: https://www.sciencedirect.com/science/article/pii/S0021999121007348.

[41] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 394 (2019) 136–152.

[42] D. Zhang, L. Lu, L. Guo, G.E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, J. Comput. Phys. 397 (2019) 108850.

[43] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. 425 (2021) 109913.

[44] D. Dubois, E. Kerre, R. Mesiar, H. Prade, Fuzzy interval analysis, in: Fundamentals of Fuzzy Sets, Springer, 2000, pp. 483–581.

[45] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, MIT press Cambridge, 2016.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[47] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[48] M.S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, G.N. Wells, The fenics project version 1.5, Arch. Numer. Softw. 3 (100) (2015) http://dx.doi.org/10.11588/ans.2015.100.20553.