Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2025) 207-221

ECCOMAS
Proceedia

UNCECOMP 2025
6th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15-18 June 2025

PHYSICS-INFORMED NEURAL NETWORKS COMBINED WITH POLYNOMIAL CHAOS EXPANSION TO PROPAGATE RANDOM FIELD PROPERTIES OF COMPOSITES

Damien Bonnet-Eymard[1], Augustin Persoons[2], Panagiotis Gavallas[3], Matthias G. R. Faes[4], George Stefanou[3] and David Moens[1]

[1] KU Leuven, Department of Mechanical Engineering,
Jan Pieter de Nayerlaan 5, 2860 Sint-Katelijne-Waver, Belgium
damien.bonnet-eymard@kuleuven.be, david.moens@kuleuven.be

[2] Université de Technologie de Troyes, Laboratory of Mechanical & Material Engineering (LASMIS),
12 rue Marie Curie- CS 42060 10010 TROYES Cedex, France
augustin.persoons@utt.fr

[3] Aristotle University of Thessaloniki, Department of Civil Engineering,
54124 Thessaloniki, Greece
pgavallas@civil.auth.gr, gstefanou@civil.auth.gr

TU Dortmund University, Chair for Reliability Engineering,
Leonhard-Euler-Strasse 5, 44227 Dortmund, Germany
matthias.faes [at] tu-dortmund[dot] de

Abstract. *Composite materials exhibit inherent structural uncertainties, particularly in the spatial distribution of their constituents, which leads to randomness in their mechanical response. If a known random variable represents the spatial distribution, a common approach to uncertainty propagation is Monte Carlo simulation. This involves computing the response for numerous samples of the random variable to approximate the output distribution. However, this method is often computationally prohibitive due to the high cost of evaluating the response for each sample. To address this challenge, surrogate models are frequently used to establish the relationship between input and output random variables. Among these, Polynomial Chaos Expansion (PCE) is a widely adopted technique, as it significantly reduces computational costs while maintaining high accuracy. However, conventional PCE frameworks are designed for problems with single output variables, making them unsuitable for predicting spatially dependent random outputs without further adaptation. This work uses a novel framework combining Physics-Informed Neural Networks (PINNs) with PCE to approximate space-dependent random outputs. This approach requires only the governing partial differential equations (PDEs) and associated boundary/initial conditions, eliminating the need for expensive forward model evaluations such as Finite Element Models. The proposed framework is applied to a composite*

Keywords: Separable Physics-Informed Neural Networks, Polynomial Chaos Expansion, Neural Chaos, Surrogate modeling

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2025.
doi: 10.7712/120225.12351.21507

plate under a uniform side load, where the stiffness coefficients are treated as random fields. Our results demonstrate the successful propagation of these random properties to predict the material's uncertain mechanical response.

1 INTRODUCTION

Due to their complex microstructure, composite materials exhibit inherent spatially varying properties, which can introduce randomness in their mechanical response. This uncertainty must be quantifie to ensure reliable design and analysis. By studying the microstructure, it is possible to calibrate random variables that represent the spatially varying material properties [1]. Here, we address the challenge of propagating this microscale uncertainty to the macroscopic mechanical response of the material.

If a model is available to compute the response of the material (e.g., a Finite Element Model), the most straightforward approach is to use Monte Carlo simulation, i.e., to compute the response for a large number of samples of the random variables in order to approximate the output distribution. This approach, nonetheless, is often computationally prohibitive, motivating the use of surrogate models to establish the relationship between the input and output random variables.

One of the most widely used surrogate modeling techniques is Polynomial Chaos Expansion (PCE) [2], which allows for a significan reduction in the number of model evaluations while maintaining high accuracy. The main idea is to approximate the model's output as a polynomial expansion of the input random variables.

The standard PCE framework is designed for problems with a single output variable and is not directly applicable to predict spatially dependent random outputs, like the displacement fiel of a composite material. A possible solution is to discretize the output fiel into a finit number of parameters (e.g., using Principal Component Analysis) and to use a separate PCE for each parameter. Another solution employs a PCE with spatially dependent coefficients removing the need for discretization approximations. The challenge then becomes determining these coefficien functions, which can no longer be fitte using regression techniques as done with scalar PCE coefficients

An interesting approach to solve this problem was proposed by Zhang et al. (2019) [3], who combined the PCE framework with Physics-Informed Neural Networks (PINNs) [4]. A neural network is used to learn the spatially dependent PCE coefficient while enforcing the governing PDEs and boundary conditions through a physics-informed loss function. It is important to note that this approach no longer requires the use of FEM simulations, as the PINN is directly used as the physics solver. In previous work [5], we replicated the results of Zhang et al. (2019) on a simple 1D Poisson equation; nonetheless, we faced difficultie scaling the method to a more complex 2D composite plate problem. The slow convergence of the PINN training, combined with the high computational cost due to the number of spatial coefficient to be learned, made the method intractable for this problem.

Indeed, for the 2D composite plate problem, approximatly 50 random variables are needed to represent the 2 spatially dependent random parameters. This leads to a total of $2 \times 50 + 2 = 102$ coefficien functions to be learned, considering only the first-orde PCE. Although first order might suffic for that problem, higher-order PCEs are often needed for a more accurate approximation. This high dimensionality is a common difficult in PCE; techniques can be used to reduce the number of basis functions to obtain a sparse PCE expansion [6].

To reduce the computational cost using sparsity, the present work combines two recently developed techniques: Neural Chaos [7] and Separable Physics-Informed Neural Networks (SPINN) [8]. Neural Chaos is a recent framework named by analogy with Polynomial Chaos Expansion, which uses a neural network instead of polynomials to construct the expansion. The random variables networks are trained alongside the coefficien functions, allowing for both

sparse basis functions and corresponding coefficient to be learned simultaneously.

A key feature of PCE is the orthogonality of its basis functions, which allows the analytical computation of output distribution moments. This orthogonality is therefore also desirable for the neural basis functions. This is achieved here using a specifi loss function that penalizes the correlation between the neural basis functions. Additionally, the SPINN architecture is used to reduce the computational cost of the training. The key idea behind SPINN is to use a separate neural network for each input dimension, which are then combined through a generalized dot product. This allows faster training and fit naturally with the random variables expansion, as detailed in Section 2.4.

Hence, the overall framework can be summarized as a combination of Separable PINNs with Neural Chaos (referred to as SPINN-NC hereafter). The SPINN-NC architecture will be detailed in the methodology section by gradually introducing each component. A brief description of the PINN and SPINN architectures will be followed by Polynomial and Neural Chaos Expansions. The combination of these two components will then be presented, leading to the SPINN-NC framework. In the second part of the paper, we firs validate the SPINN-NC approach on a simple 1D Poisson equation problem, and then apply SPINN with polynomial basis (SPINN-PC) to propagate the uncertain composite material properties.

2 Methodology

2.1 (Separable) Physics-Informed Neural Networks

The concept of utilizing neural networks to solve partial differential equations (PDEs) dates back to early works [9], later gaining significan traction through the introduction of Physics-Informed Neural Networks (PINNs) by Raissi et al. [4]. PINNs offer a framework for numerically solving boundary value problems (BVPs) by employing a neural network as the function approximator, where the network's weights and biases (θ) are the parameters to be determined.

Consider a general boundary value problem for a function $u(x)$ define on a domain Ω with boundary $\partial\Omega$:

$$\mathcal{F}_\lambda[u](x) = f(x) \quad \text{in } \Omega, \tag{1}$$

$$\mathcal{B}_\lambda[u](x) = g(x) \quad \text{on } \partial\Omega, \tag{2}$$

where x represents coordinates (potentially including time), f and g are source terms and boundary/initial conditions, and $\mathcal{F}_\lambda$ and $\mathcal{B}_\lambda$ are differential operators, possibly parameterized by λ.

PINNs approximate the solution $u(x)$ with a neural network $\hat{u}_\theta(x)$. The core idea is to integrate the governing physical laws directly into the learning process. This is achieved by definin a loss function that includes terms penalizing the extent to which $\hat{u}_\theta$ violates the PDE and the boundary conditions. These physics-based loss terms are evaluated at specifi locations called collocation points, distributed within the domain Ω and on its boundary $\partial\Omega$. If measurement data $\{(x_i, u(x_i))\}$ is available (e.g., for inverse problems to fin λ), a data fidelit term is also included. The overall loss function is typically a weighted sum:

$$\mathcal{L}_{total} = \underbrace{\alpha_{\text{data}} \sum_i \left(u(x_i) - \hat{u}_\theta(x_i)\right)^2}_{\mathcal{L}_{\text{data}}} + \underbrace{\alpha_{\text{PDE}} \sum_{x\in\Omega} \left(\mathcal{F}[\hat{u}_\theta](x) - f(x)\right)^2}_{\mathcal{L}_{\text{PDE}}} + \underbrace{\alpha_{\text{BC}} \sum_{x\in\partial\Omega} \left(\mathcal{B}[\hat{u}_\theta](x) - g(x)\right)^2}_{\mathcal{L}_{\text{BC}}}. \tag{3}$$

The hyperparameters α_i balance the contribution of each term [10]. The differential operators $\mathcal{F}$ and $\mathcal{B}$ applied to the network $\hat{u}_\theta$ are calculated efficientl using automatic differentiation (AD), a

standard feature in deep learning libraries. Boundary conditions can be enforced "softly" via the loss term $\mathcal{L}_{\text{BC}}$ or "hardly" by designing the network architecture to satisfy them by construction, which can sometimes improve convergence [4]. A schematic of the standard PINN framework is shown in Figure 1.

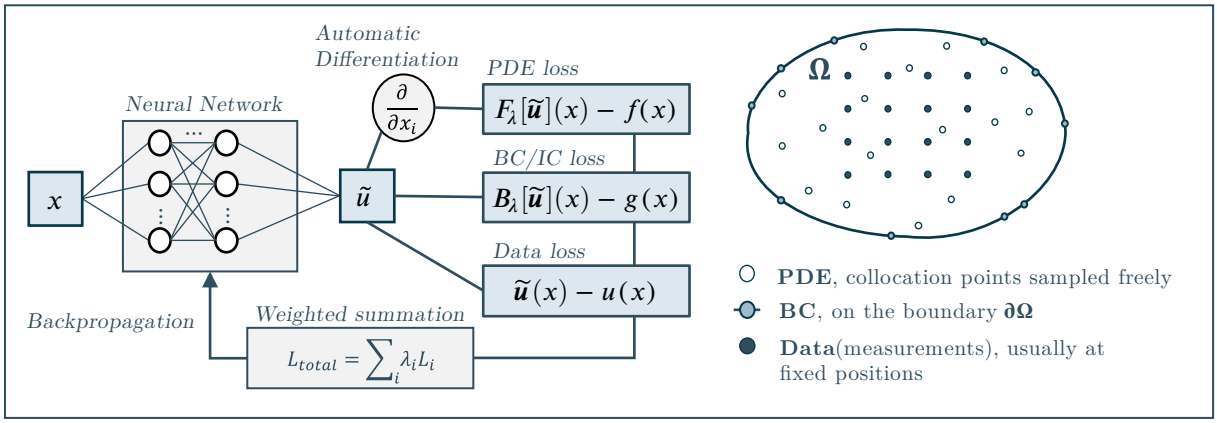

Figure 1: Physics-Informed Neural Network framework (PINN).

Training involves findin the optimal parameters θ that minimize $\mathcal{L}_{total}$, typically using gradient-based optimization algorithms. However, PINNs are known to face training challenges, and their convergence is not guaranteed, motivating ongoing research into improving their robustness and efficien y [10, 11].

To address these convergence issues, Separable Physics-Informed Neural Networks (SPINNs) were introduced by Cho et al. (2023) [8]. This architecture, illustrated in Figure 2, employs a separable structure where each component of the d-dimensional input $x = (x_1, \ldots, x_d)$ is fed into a separate "body-network". Each body-network produces an output vector of size r (the rank of the model). These d output vectors are then combined through a generalized dot product to produce the fina output $\hat{u}_\theta(x)$. This structure can be interpreted as a low-rank tensor approximation of the solution, and SPINNs possess universal approximation capabilities [8].

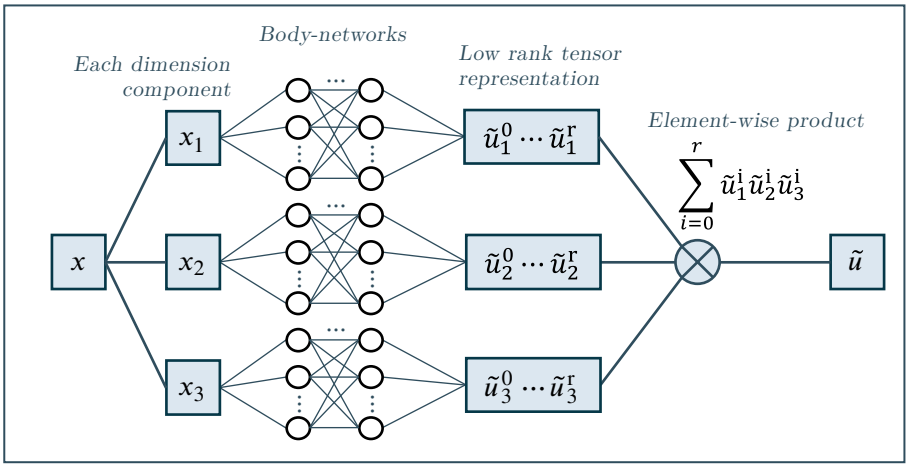

Figure 2: Separable PINN architecture for a 3D problem.

The primary advantage of SPINN lies in its computational efficien y during training. To compute the network output at N^d points arranged on a Cartesian product grid (formed by

taking N points along each of the d dimensions), only N evaluations of each of the d body-networks are required, totaling $d \times N$ evaluations. This is a significan reduction compared to the N^d evaluations needed for a standard feedforward network. This benefit which extends to gradient computations via forward-mode AD, allows the physics loss terms to be evaluated at a much larger set of collocation points for a comparable computational cost, often leading to substantially faster convergence and improved accuracy, with reported wall-clock time reductions exceeding an order of magnitude [8].

However, this efficien y comes with the constraint that evaluation points must lie on a factorizable, grid-like structure. This can complicate applications involving complex domain geometries [8], but is not restrictive for the simple domains studied here.

Furthermore, the tensor product structure inherent in the SPINN architecture, resulting from the combination of outputs from dimension-specifi networks, aligns naturally with the structure of random variables expansions.

2.2 Polynomial Chaos Expansion

Polynomial Chaos Expansion (PCE) is a popular surrogate modeling technique that has been widely used in uncertainty quantification [12] This framework will be only briefl introduced here, the interested reader is referred to the UQLab user manual [13] for more details. The goal of PCE is to propagate uncertain input parameters through the output of a model. The main idea is to represent this ouptut as a polynomial expansion of the random input parameters. Consider a model $\mathbf{Y} = \mathcal{M}(\xi)$ where $\xi \in \mathbb{R}^M$ is a random vector with independent components and $\mathbf{Y}$ is the output of the model. The PCE surrogate model of $\mathcal{M}$ is given by

$$\mathbf{Y} \approx \mathcal{M}^{\text{PCE}}(\xi) = \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\xi), \tag{4}$$

where $\mathcal{A} \subset \mathbb{N}^M$ is the set of multi-indices, y_α are the PCE coefficients and $\Psi_\alpha(\xi)$ are the multivariate polynomials. The polynomials are constructed to be orthogonal with respect to a probability distribution of the input parameters (see [13] for more details). The set of multi-indices $\mathcal{A}$ encodes the truncation of the expansion, which is a trade-off between accuracy and computational cost. Many strategies exist in the literature to optimally choose this truncation (e.g. sparse PCE [6]).

Once the PCE basis functions $\Psi_\alpha(\xi)$ are defined calibrating the model consists of findin the PCE coefficient y_α. This step is usually data-based: the model $\mathcal{M}$ is evaluated at a set of points and the PCE coefficient are determined by regression.

Classic PCE models, as written in Eq. 4, only take random variables as input. However, in our case, the space coordinates (and eventually time) should also be considered as input. There are different ways to handle space variables in the PCE framework. The main one is to make the coefficient of the PCE model dependent on the space variables: $y_\alpha = y_\alpha(x)$

2.3 Neural Chaos

Inspired by the spectral expansion formalism of PCE, Neural Chaos (NC) was originally introduced as a data-driven alternative to classical polynomial chaos methods [7]. The main idea in Neural Chaos is to replace the pre-define polynomial basis functions $\Psi_p(\xi)$ with basis functions parameterized by neural networks, $\Psi_p(\xi; \beta_p)$, where β_p represents the network parameters.

The expansion then takes a form analogous to that of PCE:

$$\mathbf{Y} \approx \sum_{i=0}^{r} y_i(x;\theta_i)\Psi_i(\xi;\beta_i), \tag{5}$$

with the deterministic coefficient $y_i(x;\theta_i)$ also represented by neural networks.

In the original implementation, calibration was performed solely using data, with the stochastic basis functions constructed sequentially through iterative algorithms to ensure orthogonality. In our approach, orthogonality is enforced by incorporating a loss term that compares the covariance matrix of the network outputs with the identity matrix, thereby maintaining uncorrelated learned stochastic basis functions $\Psi_i(\xi;\beta_i)$.

Furthermore, we depart from the data-based methodology by integrating Neural Chaos within a Physics Informed Neural Network (PINN) framework, directly leveraging the underlying physics.

2.4 PINN with Polynomial and Neural Chaos

The PINN-PC framework, proposed by Zhang et al. (2019) [3], uses a neural network to learn the spatially dependent Polynomial Chaos (PC) coefficient $y_\alpha(x)$. As depicted in Figure 3, the network receives the spatial coordinates x as input and outputs the set of coefficient $\{y_\alpha(x)\}_{\alpha\in\mathcal{A}}$. These coefficient then multiply the known PC basis functions $\{\Psi_\alpha(\xi)\}$ from Eq. 4 to form the overall approximation $u_\theta(x,\xi)$. Because the PCE expansion is simply a linear combination and the neural network is differentiable, the combined model remains differentiable with respect to x. Automatic differentiation can thus be used to compute the PDE and boundary condition residuals needed for the physics-informed loss terms. However, since one must sample not only over spatial points x but also over random variables $\{\xi_s\}_{s=1}^{N_\xi}$, the computational load can become prohibitive, as observed in our previous work [5].

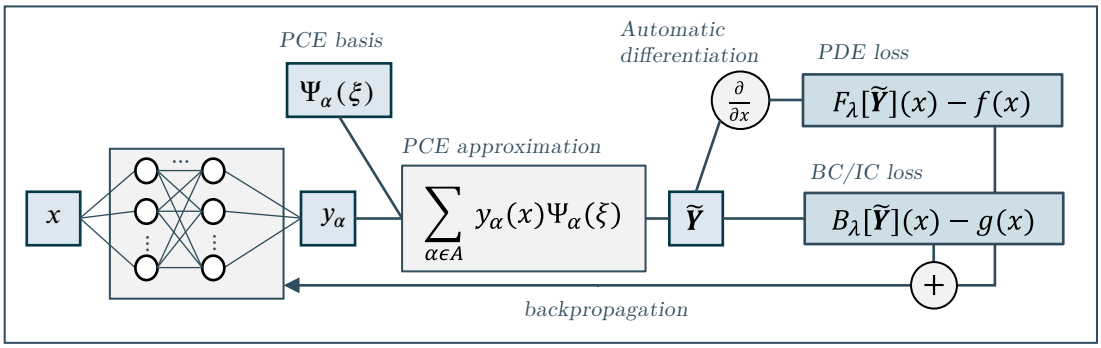

Figure 3: Physics-Informed Polynomial Chaos framework (**PINN-PC**).

To alleviate these challenges, we propose a Separable Physics-Informed Neural Chaos framework (see Figure 4). Instead of relying on fi ed PC basis functions, we use Neural Chaos (NC) to learn the random basis functions $\{\Psi_i(\xi)\}_{i=0}^{r}$ directly. Simultaneously, the SPINN architecture processes spatial coordinates x to produce corresponding deterministic coefficient $\{y_p(x)\}_{i=0}^{r}$. The fina approximation is formed by the dot product of these two sets of outputs:

$$\mathbf{Y} = \sum_{i=0}^{r} y_i(x)\,\Psi_i(\xi).$$

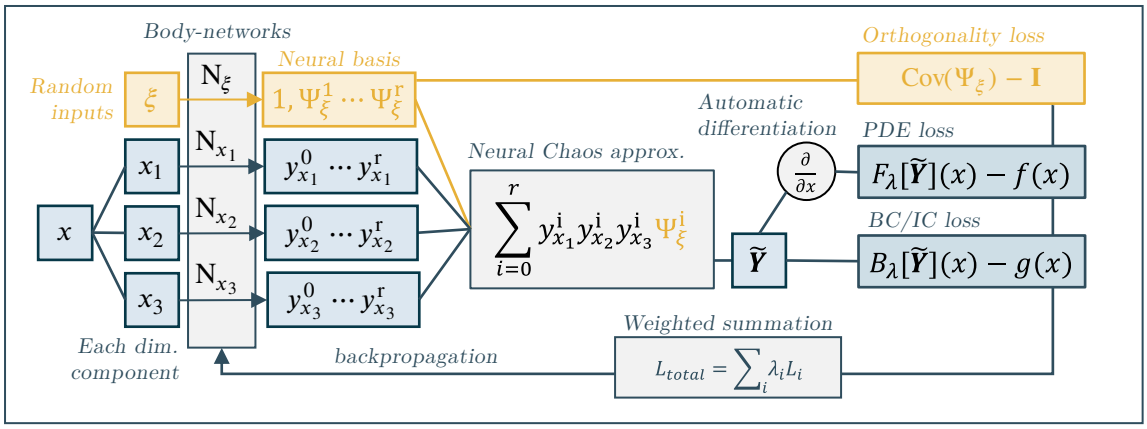

Figure 4: Separable Physics-Informed Neural Chaos framework (**SPINN-NC**).

Separable architecture. Each dimension (space or random) is handled by its own body network, improving computational efficien y when training over both spatial grids and random samples. This structure enforces separability at the level of each i-mode, potentially limiting the capacity of the model. This can be mitigate using higher-rank expansions (r).

Zero-order mode. A dedicated unit vector is included for the zero-order mode, representing the mean component of the solution. This ensures that $\Psi_0(\xi) \equiv 1$, and $y_0(x)$ captures the overall mean field

Orthogonality loss. Orthogonality among the learned random basis functions is enforced through a loss term that compares their covariance matrix (computed over a batch of ξ_s) to the identity matrix. To ensure these basis functions remain centered (i.e., orthogonal to the mean mode), no bias is used in the random body network, and an odd activation function (e.g., $\tanh$) is selected.

By unifying Neural Chaos for the random dimension with a SPINN-based approach for the spatial dimension, we can expect significan gains in computational speed compared to traditional PINN-PC. Coefficient predicted by a SPINN can also be combined with a polynomial basis, corresponding to a model called SPINN-PC hereafter.

3 Application

In this section, we firs validate the Neural Chaos approach on a simple 1D Poisson equation problem. We then address the propagation of uncertain composite plate material properties. All the simulations are performed using the open-source library `DeepXDE` [14], on a single NVIDIA GeForce RTX 4080 Ti GPU. The code will be made available on GitHub: `www.github.com/bonneted/PINN-PC`.

3.1 1D Poisson Equation

For initial validation, we reproduce the results from Zhang et al. [3] using Neural Chaos instead of Polynomial Chaos.

3.1.1 Problem Statement

We consider the one-dimensional stochastic Poisson equation with Dirichlet boundary conditions:

$$\begin{aligned} -\frac{d^2u}{dx^2} &= f(x,\xi), \quad x \in [0,1], \\ u(0) &= u(1) = 0. \end{aligned} \tag{6}$$

The source term $f(x,\xi)$ is modeled as a Gaussian random field, $\mathcal{GP}(f_0(x), k(x,x'))$, with a mean function given by

$$f_0(x) = 10\sin(2\pi x),$$

and a squared exponential covariance:

$$k(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{l^2}\right), \tag{7}$$

with $\sigma = 1$ and $l = 0.5$. To represent the randomness, $f(x,\xi)$ is discretized using a Karhunen-Loève expansion with $N_\xi = 6$ terms, retaining 99% of the variance.

3.1.2 Results

The SPINN-NC method is trained using $N_x = 13$ collocation points (denoted as f sensor on Figure 5 as it corresponds to points where the forcing term f is known) and $N_\xi = 10\,000$ samples of the random variables. We choose a rank $r = 3$ which is half the number of random variables. Training is carried out with the Adam optimizer at a learning rate of 10^{-3} for 5 000 iterations (approximately one minute). Orthogonality of the neural basis functions is enforced during training, ensuring that the firs coefficien provided by the model represents the mean of the solution and the subsequent coefficient capture the uncertainties.

The covariance matrix, M_{cov}, of the basis functions at the end of the training is approximated using 10^5 samples of the random variables:

$$M_{cov} = \begin{bmatrix} 0.991 & 0.006 & -0.000 \\ 0.006 & 1.002 & -0.006 \\ -0.000 & -0.006 & 1.004 \end{bmatrix} \tag{8}$$

As it is close to the identity matrix, the mean and standard deviation of the solution can directly be computed from the coefficient of the Neural Chaos expansion. They are compared to the reference solution obtained from Monte Carlo simulations in Figure 5.

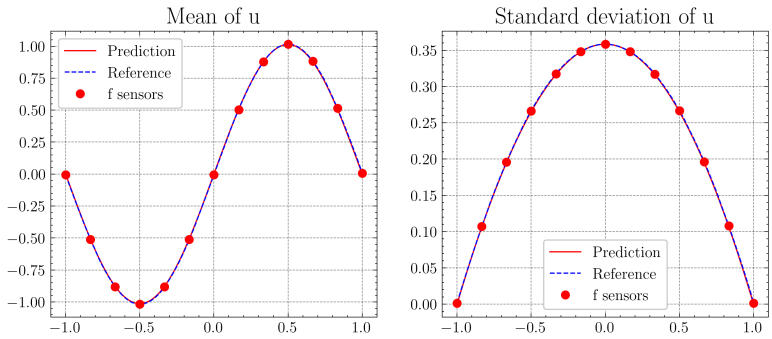

Figure 5: Mean and standard deviation of the SPINN-NC approximation compared with Monte Carlo reference solution (10^4 samples)

To further validate the accuracy of the approximation, we computed the mean squared error (MSE) between the Neural Chaos solution and a reference solution obtained using finit difference. This comparison was performed on a set of samples used during training (Train) and on a new, independently sampled set (Test) to assess potential overfitting The density plots illustrating these MSE distributions are presented in Figure 6. The resulting mean squared error for the Neural Chaos solution was consistently below (10^{-3}) and are very similar between the training and test datasets. This indicates an accurate approximation, even with a rank lower than the number of random variables, thereby encouraging the further exploration of Neural Chaos for dimension reduction.

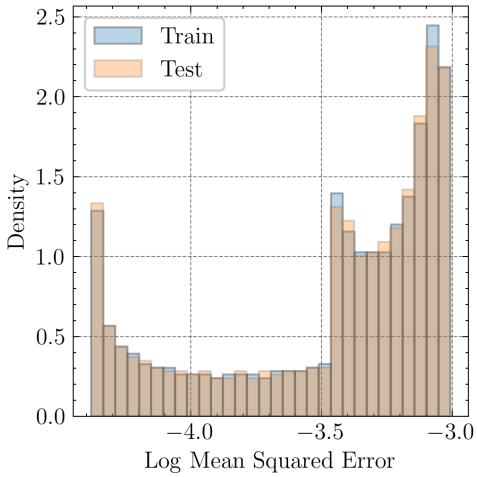

Figure 6: Density plots of the mean square error of the SPINN-NC approximation.

3.2 Composite plate

Building upon the work of Stefanou et al. (2022) [1], we investigate the propagation of random spatial variations in the stiffness matrix components of a composite plate, where these components are modeled as random field calibrated from microstructure simulations.

3.2.1 Problem statement

We consider a 2D composite plate of dimensions 100μm $\times$ 100μm. A tensile side load $p = 0.1\text{N}/\mu\text{m}$ is applied on the right side of the plate. The boundary conditions are illustrated

in Figure 7.

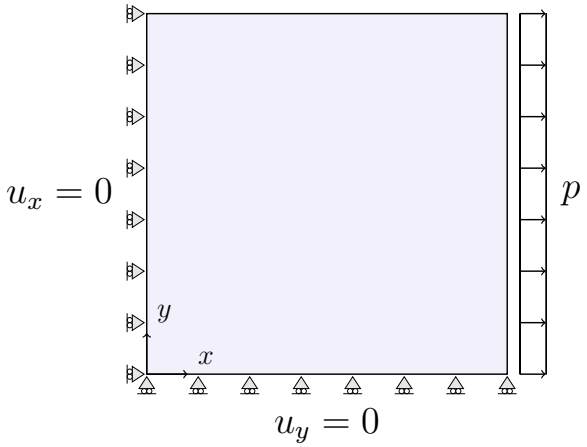

Figure 7: Boundary value problem for the composite plate.

The material is considered to be linear elastic and isotropic. The stiffness matrix can be expressed in terms of stiffness components C_{11} (axial) and C_{33} (shear) as follows:

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{11} - 2C_{33} & 0 \\ C_{11} - 2C_{33} & C_{11} & 0 \\ 0 & 0 & C_{33} \end{bmatrix}. \tag{9}$$

The stiffness components C_{11} and C_{33} are modeled as lognormal random field with realistic mean and standard deviation calibrated from simulations of the microstructure, after Stefanou et al. (2022) [1]. To limit the number of terms in the KL expansion, we firs consider larger values for the correlation length L_x and L_y (in μm). The parameters are summarized in Table 1.

Table 1: Parameters of the underlying Gaussian random field modeling the stiffness components.

Parameter	**Mean**	**Std**	**Lx**	**Ly**
C_{11}	1.55	0.129	75	100
C_{33}	0.316	0.027	100	75

For illustration, a realization of both C_{11} and C_{33} random field is shown in Figure 8.

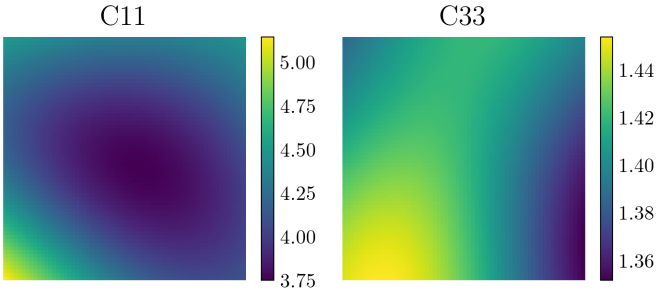

Figure 8: Realization of the random field C_{11} and C_{33}.

Similarly to the Poisson equation example, the random field are discretized using the Karhunen-Loève expansion. We choose here a squared exponential covariance function for the random field to keep the number of basis functions low. The problem is discretized on a grid of $N = 100 \times 100$ points. Eigendecomposition of the covariance matrix is performed, and truncation of the expansion is chosen to keep 95% of the variance, resulting in $M = 4$ basis functions for each stiffness component.

3.2.2 Results

On this first attempt with larger correlation lengths, the number of random variables is small enough to directly use them as basis functions (no dimension reduction with Neural Chaos). The spatial coefficient are approximated using a Separable-PINN and the SPINN-PC model is trained using $N_x = 100^2$ collocation points and $N_\xi = 10$ samples of the random variables. The training is performed with the Adam optimizer at a learning rate of 10^{-3} for 50 000 iterations (approximately 5 minutes). The mean and standard deviation of the displacement field are computed by sampling the random variables (Figure 10). The results are compared to the reference solution obtained from 10^3 Monte Carlo simulations shown in Figure 9.

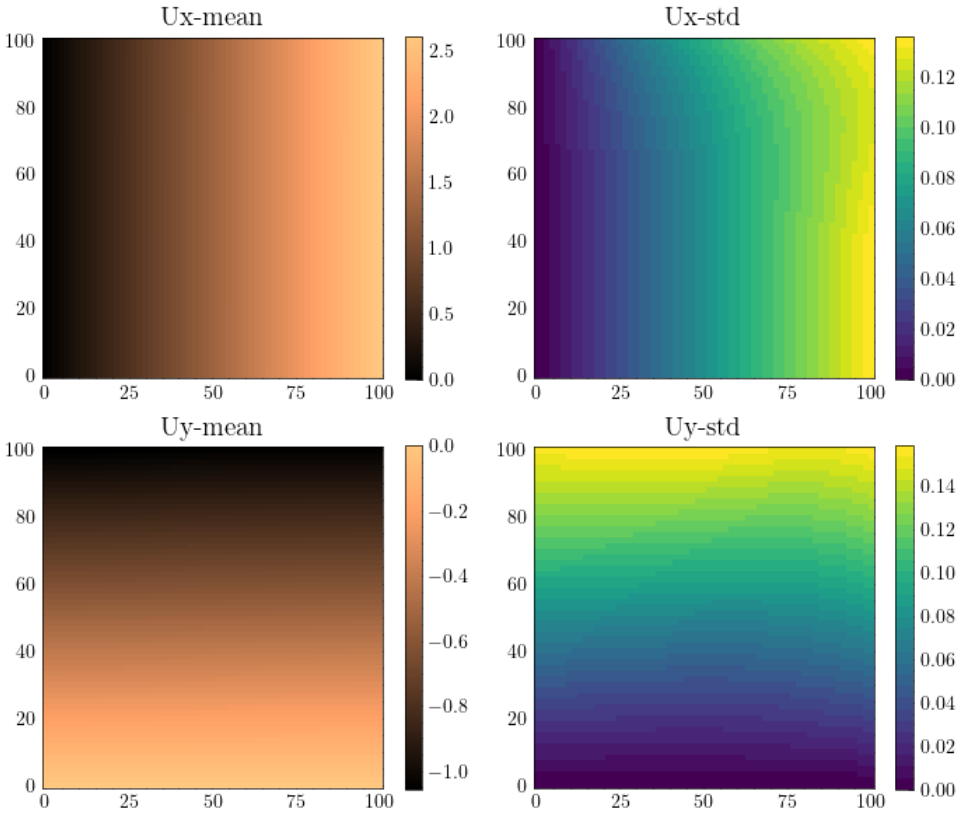

Figure 9: Mean and standard deviation of the displacement field for the composite plate problem using the 10^3 Monte Carlo samples.

The mean and standard deviation of the displacement field are found to be in good agreement with the reference solution, validating the use of Physics-Informed Neural Networks combined with Polynomial Chaos Expansion to propagate the random properties of the composite material.

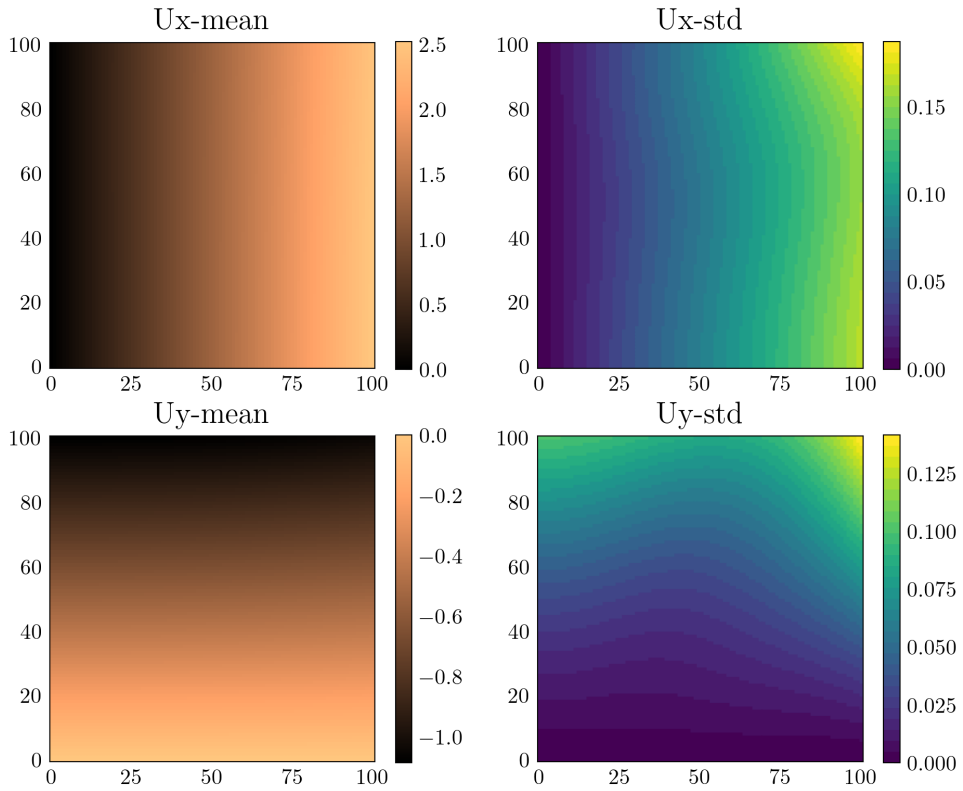

Figure 10: Mean and standard deviation of the displacement fiel for the composite plate problem using the SPINN-PC method.

The next step consists of using realistic correlation lengths for the random fields This will increase the number of random variables to approximately 50, making the problem intractable with direct polynomial expansion. To address this, we intend to leverage the Neural Chaos framework for dimensionality reduction. While the initial SPINN-PC results are promising, the full implementation of SPINN-NC for this more complex problem remains under development.

4 Conclusion

In this work, we proposed a novel framework combining Separable Physics-Informed Neural Networks (SPINN) with Neural Chaos (NC) to propagate the random properties of composite materials. The SPINN-NC architecture allows for efficien training over both spatial and random dimensions, leveraging the separable structure of SPINN while learning the random basis functions directly through NC. The method was validated on a simple 1D Poisson equation problem, demonstrating its accuracy and efficien y. Subsequently, we applied a SPINN model with a polynomial basis (SPINN-PC) to a composite plate problem with larger correlation lengths, successfully validating the use of Physics-Informed Neural Networks for this class of problems. Our next step involves applying the full SPINN-NC framework to a more realistic scenario with smaller correlation lengths, leveraging the Neural Chaos framework for dimensionality reduction

REFERENCES

[1] G. Stefanou, D. Savvas, P. Gavallas, and I. Papaioannou, "The effect of random fiel parameter uncertainty on the response variability of composite structures," *Composites Part C: Open Access*, vol. 9, p. 100324, Oct. 2022.

[2] B. Sudret, "Polynomial chaos expansions and stochastic finit element methods," in *Risk and Reliability in Geotechnical Engineering* (J. C. Kok-Kwang Phoon, ed.), pp. 265–300, CRC Press, 2015.

[3] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," *Journal of Computational Physics*, vol. 397, p. 108850, Nov. 2019.

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

[5] D. Bonnet Eymard, A. Persoons, P. Gavallas, D. Moens, M. G. Faes, and G. Stefanou, "Physics-Informed Neural Networks to propagate random fiel properties of composite materials," in *Proceedings of ISMA2024 International Conference on Noise and Vibration Engineering USD2024 International Conference on Uncertainty in Structural Dynamics*, KU Leuven, Departement Werktuigkunde, LMSD (Mecha(tro)nic System Dynamics), Oct. 2024.

[6] G. Blatman and B. Sudret, "Adaptive sparse polynomial chaos expansion based on least angle regression," *Journal of Computational Physics*, vol. 230, pp. 2345–2367, Mar. 2011.

[7] B. Bahmani, I. G. Kevrekidis, and M. D. Shields, "Neural Chaos: A Spectral Stochastic Neural Operator," Feb. 2025. arXiv:2502.11835 [cs].

[8] J. Cho, S. Nam, H. Yang, S.-B. Yun, Y. Hong, and E. Park, "Separable Physics-Informed Neural Networks," Oct. 2023. arXiv:2306.15969 [cs].

[9] I. Lagaris, A. Likas, and D. Fotiadis, "Artificia neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, pp. 987–1000, Sept. 1998. Conference Name: IEEE Transactions on Neural Networks.

[10] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, *An Expert's Guide to Training Physics-informed Neural Networks*. Aug. 2023.

[11] S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopulos, and G. E. Karniadakis, "Learning in PINNs: Phase transition, total diffusion, and generalization," Mar. 2024. arXiv:2403.18494 [cs].

[12] S. Sakamoto and R. Ghanem, "Polynomial Chaos Decomposition for the Simulation of Non-Gaussian Nonstationary Stochastic Processes," *Journal of Engineering Mechanics*, vol. 128, pp. 190–201, Feb. 2002. Publisher: American Society of Civil Engineers.

[13] S. Marelli and B. Sudret, *UQLab user manual - Polynomial chaos expansions*. July 2015.

[14] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *SIAM Review*, vol. 63, pp. 208–228, Jan. 2021. arXiv: 1907.04502.