POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Physics-informed neural networks for detecting defects in variable angle tow laminates

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

Author: **Giuseppe Emanuele Tabbò**

Student ID: 966013
Advisor: Riccardo Vescovini
Co-advisors: Cheng Angelo Yan
Academic Year: 2021-22

Abstract

Variable-Angle Tow (VAT) composites are becoming more and more popular for engineering structural purposes. This is due to the possibility of designing structural load paths by steering the reinforcing fibers along curvilinear paths, instead of keeping them straight. Unfortunately, their manufacturing processes, the Automated Fiber Placement (AFP) and Continuous Tows Shearing (CTS), are source of different types of defects like gaps, overlaps, fibers misalignment and out-of-plane waviness. The latter can cause large variations of the mechanical performances of the structures from the nominal ones.
For this reason, a method for the detection and quantification of the defects mentioned above is developed. In particular, the Physics-Informed Neural Networks (PINNs) and a tool commonly used for the representation of stochastic processes, the Karhunen-Loeve expansion (KLE), are exploited.
Moreover, the effectiveness of the proposed technique is tested by tackling different problems, starting from the detection of Young modulus for a rod, until the detection of gaps, overlaps and fibers misalignment for a VAT laminate.

Keywords: Variable-Angle Tow composites, Automated Fiber Placement, Continuous Tow Shearing, Physics-Informed Neural Network, Karhunen-Loeve expansion

Abstract in lingua italiana

I compositi a rigidezza variabile (VAT) stanno diventando sempre più popolari in ambito ingegneristico strutturale. Questo è dovuto alla possibilità di realizzare percorsi di carico strutturale ponendo le fibre lungo percorsi curvilinei, invece di mantenerle rettilinee. Sfortunatamente, i relativi processi di produzione, l'Automated Fiber Placement (AFP) ed il Continuous Tow Shearing (CTS), sono fonte di diversi tipi di difetti, quali lacune, sovrapposizioni e disallineamento di fibre. Quest'ultimi possono causare un forte deterioramento delle proprietà meccaniche delle strutture.
Per questa ragione, un metodo per il rilevamento e la quantificazione dei difetti dei compositi VAT è sviluppato. In particolare, le reti neurali del tipo Physics-informed e uno strumento comunemente utilizzato per la rappresentazione di processi stocastici, l'espansione di Karhunen-Loeve, sono utilizzati a tale scopo.
Inoltre, l'efficacia del metodo proposto è testata attraverso la soluzione di diversi problemi, a partire dalla quantificazione del modulo di Young per una trave, fino al rilevamento dei difetti per un laminato VAT.

Parole chiave: Strutture a rigidezza variabile, Automated Fiber Placement, Countinuous Tow Shearing, PINNs, espansione di Karhunen-Loeve

Contents

1 | Introduction

Manufacturing processes are responsible of defects arising inside the structures, which can cause large variations of mechanical performances with respect to the expected ones. In this regard, industries always look for improving quality, reliability and robustness of the structures thanks to technology development in manufacturing machinery. However, some kinds of defects are still unavoidable because they are intrinsic of the manufacturing processes.
Their effect on mechanical performances can be quantified by means of the Sobol indices [41], which are coefficients computed during the sensitivity analyses. In particular, the latter exploit the so-called surrogate models [20, 39], like Polynomial Chaos Expansion (PCE) [45] and Karhunen Loeve Expansion (KLE) [43], to speed-up this procedure.
Even if the sensitivity analysis represents a very useful tool, it is not able to return the exact value of the structural output of interest, but only its probabilistic features (average value and variance). This leads to use the safety coefficients, which are very limiting from the structural design point of view.
For this reason, the development of methods able to detect the defects in good approximation and to model them numerically is becoming a widespread activity.

One of the most used material in aeronautical industry is the composite. The manufacturing process of a composite structure is lengthy and complex, such that it could be the origin of different types of defects. The latter can be the consequence of many factors, like mould selection and bagging configuration [17].
Particularly in the last decades, a new variant of composite structures gained more and more success: the Variable-Angle Tow (VAT) structures. The latter are characterized by reinforcing fibers steered along a curvilinear path, instead of keeping them straight. This allows for the design of structural load paths useful to increase performances and to provide more flexibility for trade-offs between structural properties.
In this regard, Lincoln et al. [26] found that variable stiffness laminates could simultaneously maximize two conflicting structural design objectives: the buckling load, one of the most important requirement in aeronautics, and the in-plane stiffness; in [34], instead, it

is shown that an open hole variable stiffness plate, adequately designed, possessed better post-buckling performances, higher buckling load and ultimate strength with respect to a constant stiffness design.
The main drawback of VAT structures relates with their manufacturing techniques, Automated Fibre Placement (AFP) and Continuous Tows Shearing (CTS) [23], which are both source of many defects like fibre waviness, gaps and overlaps [22]. In this regard, many experimental studies were done to quantify their effects on the mechanical performances of the structures like in-plane stiffness, buckling load and, in particular, strength [15, 46]. In the latter, composite laminates with intentionally placed tow-to-tow gaps and overlaps, manufactured by means of AFP, were tested in tension and compression in order to compare the results obtained with pristine laminates. It resulted that the most affected property was the strength, because of the out-of-plane fiber waviness. In particular, the latter is the consequence of the presence of gaps and overlaps and promotes shear interlaminar stress and out-of-plane normal stress, which are responsible of the earlier failure. Since experimental analysis could be lengthy and costly, progresses were done regarding the numerical modelling of the defects. In [4, 13], starting from the knowledge of defects location, it was illustrated how to modify the properties of a 2D finite elements model to take into account the presence of gaps and overlaps. Concerning the modelling of the out-of-plane waviness, which arises after the compaction phase as a consequence of the presence of gaps and overlaps, it is necessary to realize 3D finite element models which are able to reproduce the waviness observed through optical images of the laminate under investigation, as illustrated in [25, 32].
These modelling strategies can be employed to build finite element models of the real defected structures in order to make faster and cheaper sensitivity analyses with respect to the experimental ones [13, 30].
It is obvious that before building the finite elements model, it is necessary to know the position of the defects inside the structure. Usually, this is done by means of non-destructive techniques, like radiography and thermography, which require specialized machinery and skilled personnel for the interpretation of the results [10].

In the last years another instrument was employed in the context of defects detection: the Artificial Neural Networks (ANNs).
These are mathematical models that, within the framework of structural mechanics, are widely used to solve forward problems. In [28, 35] the ANNs were exploited to quantify the quality of the weld bead using the acoustic signals recorded during the welding process as inputs of the network, while in [27] for detecting cracks affecting aluminum

plates exploiting the information regarding the propagation of Lamb waves. Regarding composite structures, in [8] the ability of ANNs in solving non-linear problems was used together with Thermographic Non Destructive Evaluation (TNDE), not only for the detection of the defects, but also for the estimation of their depth through the thickness of the laminate. In this last case the inputs of the network were temperature versus time curves.
These techniques are faster and cheaper with respect to the typical non-destructive techniques and, moreover, the network outputs are easy to be interpreted.
Sometimes the quantity of labeled data available could be limited, such that the ANN can hardly be trained. In this case the network must be enriched by means of other known information, like the physical model describing the behaviour of the system under investigation. This variant of the neural network goes under the name of Physics-Informed Neural Network (PINN) and it can be used for the solution of both forward and inverse problems. In [37] and [16], the PINNs were employed to solve forward problems, like the solution of the one-dimensional Schrôdinger equation and the linear static analysis of a plate.
PINNs are also employed for the solution of inverse problems, which consists in the detection of unknown parameters regarding the underlying model of the system considered. In [16] the PINN was used to identify the Lamè parameters (λ, μ) defining the constitutive model of a material, while in [49] the identification of VAT structures natural frequencies and laminae fibers path parameters is pursued.

Therefore, in this thesis it is proposed a non-intrusive method for the detection of both location and entity of different types of defects affecting VAT structures. In particular, it exploits the ability of the PINNs in solving inverse problems and the possibility to represent the defects distributions in terms of stochastic fields using the Karhunen-Loeve expansions.
Differently from the techniques employing the neural networks cited above, the training process needs easily accessible quantities, like the displacements and strains fields. Moreover, the outputs of the network are structured in a way they can be used for the generation of numerical models of the defected structures.
In order to illustrate the development of this method and the way it works, the thesis is organized as follows:

- Chapter 2 presents some basic elements about composite structures and a description about VAT structures designs, manufacturing processes and defects modelling;
- Chapter 3 is focused on sensitivity analysis and in particular on the surrogate models

used to speed up them;

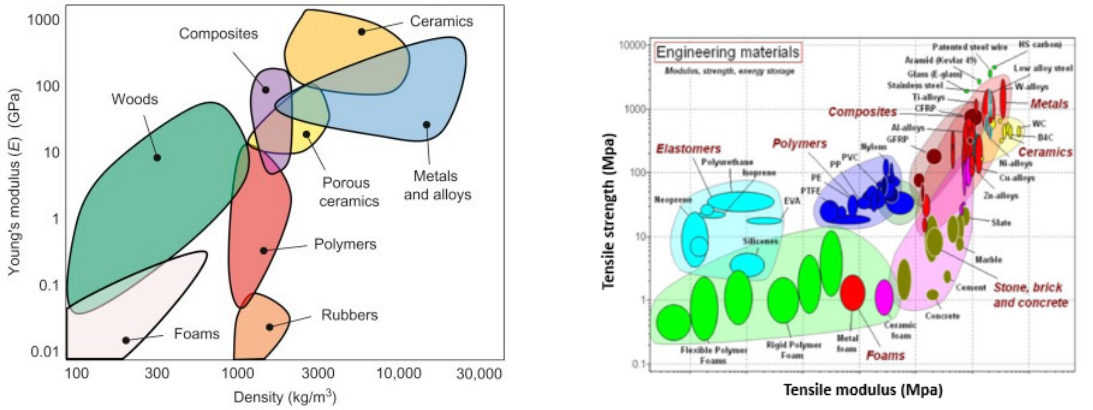
- Chapter 4 presents the neural networks, illustrating the variants (ANN and PINN) and the training methods;
- Chapter 5 explains in detail the developed method for the detection of defects;
- Chapter 6 presents the results obtained during the validation phase;
- Chapter 7 for the conclusions and future developments.

2 | Composite structures

In this chapter, basic elements regarding the composite material, the *'Rule of Mixtures'* and *'Lamination Theory'*, are recalled at the beginning. Then, the Variable-Angle Tow structures are introduced focusing on possible fibers path design, manufacturing techniques along with the related defects and on the numerical modelling techniques for the defects.

2.1. Introduction to composite material

The composite is a bi-phase material made of constituents characterized by significantly different physical and chemical properties, which are combined in order to obtain a material that exploits their advantageous properties.
In particular, when dealing with aircraft structures, composite is intended as made by a reinforcement, the fibers, characterised by high mechanical performances which is embedded in a matrix, a polymer, that has the purpose to hold the reinforcement in a desired shape. This material gained success in the last decades because of its excellent mechanical properties associated to a low density, as shown in Ashby diagrams [40] in Figure 2.1.



(a) Young Modulus vs.Density [21]

(b) Tensile Strength vs.Tensile modulus [40]

Figure 2.1: Ashby diagrams

One of the most representative index that can be derived from the diagrams above is the specific modulus RTM, which gives the idea about the weight of a structure with a certain stiffness. It is possible to find out that its value is about 2.5 for all metals, while for the composite it can reach the value of 10 (i.e. for the carbon fibers reinforced one). This shows that the composite allows to save a lot of weight, one of the most important requirements in aeronautics, without losing in mechanical performances.

2.1.1. In-plane composite material elastic properties

The structures made of composite employed in aeronautics are often thin-walled, which means that, in several circumstances, the plane stress state is a reasonable assumption. For this reason, this section is focused on how obtaining the in-plane elastic properties. The elastic properties of a composite material depends on the type of reinforcing fibers (carbon, glass, Kevlar and others) , matrix (i.e. epoxy resin) employed and on their volumetric fractions (V_m for the matrix and V_f for the fibers). These data are used in the *'Rule of Mixtures'* as follows:

$$\begin{cases} E_{11} = E_f V_f + E_m V_m, \\ E_{22} = \dfrac{E_m E_f}{E_m V_f + E_f V_m}, \\ \nu_{12} = \nu_{12f} V_f + \nu_{12m} V_m, \\ G_{12} = \dfrac{G_{12m} V_f}{G_{12m} V_f + G_{12f} V_m}. \end{cases} \tag{2.1}$$

where the subscript 1 is used to denote fibers direction and the subscript 2 the direction perpendicular to the former, as shown in Figure 2.2.

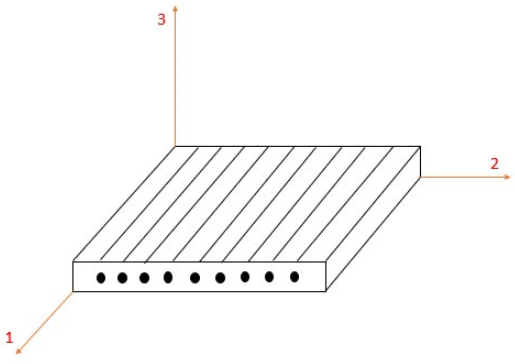


Figure 2.2: Lamina reference system

Subsequently, Eq. (2.1) can be exploited to write the stiffness matrix $\hat{\boldsymbol{D}}$ of an unidirectional reinforced composite as follows:

$$\hat{\boldsymbol{D}} = \begin{bmatrix} \frac{E_{11}}{1-\nu_{12}\nu_{21}} & \frac{E_{11}\nu_{12}}{1-\nu_{12}\nu_{21}} & 0 \\ \frac{E_{22}\nu_{21}}{1-\nu_{12}\nu_{21}} & \frac{E_{22}}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{2.2}$$

In particular, $\boldsymbol{\sigma} = \hat{\boldsymbol{D}}\boldsymbol{\epsilon}$, where $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{22}, \tau_{12})^T$ and $\boldsymbol{\epsilon} = (\epsilon_{11}, \epsilon_{22}, \gamma_{12})^T$.

2.1.2. Lamination Theory

A plate-shaped composite structure is a set of a laminae stacked up together. Number of laminae, lamina reinforcing fibers orientation and position through the thickness are chosen in order to obtain the desired final shape and mechanical performances.

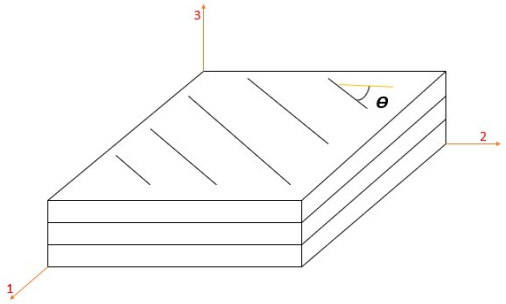


Figure 2.3: Laminate reference system

The elastic properties of the laminate depend on those of each single lamina, their fibers orientation θ with respect to the laminate reference system (see Figure 2.3) and their position through the laminate thickness, as stated in the *'Lamination theory'*.
The latter allows to find a relation between the forces $\boldsymbol{N}$ and moments $\boldsymbol{M}$ per unit length, defined as follows:

$$\begin{cases} \boldsymbol{N} = \int_{-t/2}^{t/2} \boldsymbol{\sigma} dz, \\ \boldsymbol{M} = \int_{-t/2}^{t/2} \boldsymbol{\sigma} z dz, \end{cases} \tag{2.3}$$

and the generalized deformations vector $\boldsymbol{\epsilon} = (\epsilon_x, \epsilon_y, \gamma_{xy}, k_x, k_y, k_{xy})^T$.
In particular, the laminate stiffness matrix $\hat{\boldsymbol{D}}_{\boldsymbol{L}}$, such that $(\boldsymbol{N}, \boldsymbol{M})' = \hat{\boldsymbol{D}}_{\boldsymbol{L}}\boldsymbol{\epsilon}$, has the following form:

$$\hat{\boldsymbol{D}}_{\boldsymbol{L}} = \begin{bmatrix} \hat{\boldsymbol{A}} & \hat{\boldsymbol{B}} \\ \hat{\boldsymbol{B}} & \hat{\boldsymbol{D}} \end{bmatrix} \tag{2.4}$$

Assuming that each lamina has different elastic properties, but constant through its thickness, the matrices in Eq. (2.4) can be can be written as follows:

$$\begin{cases} \hat{\boldsymbol{A}} = \sum_{i=1}^{n} \hat{\boldsymbol{D}}_i (h_i - h_{i-1}), \\ \hat{\boldsymbol{B}} = \frac{1}{2} \sum_{i=1}^{n} \hat{\boldsymbol{D}}_i (h_i^2 - h_{i-1}^2), \\ \hat{\boldsymbol{D}} = \frac{1}{3} \sum_{i=1}^{n} \hat{\boldsymbol{D}}_i (h_i^3 - h_{i-1}^3), \end{cases} \tag{2.5}$$

where n is the number of laminae, $(h_i; h_{i-1})$ are the through the thickness coordinates of i-th lamina faces and $\hat{\boldsymbol{D}}_{\boldsymbol{i}}$ is the stiffness matrix of the i-th lamina rotated in the laminate reference system.
Denoting with $\hat{\boldsymbol{D}}_{\boldsymbol{li}}$ the stiffness matrix in the lamina reference system (see Figure 2.2) and assuming that the lamina is rotated by an angle θ (clockwise direction) around the third axis of the laminate reference system, it can be found that $\hat{\boldsymbol{D}}_{\boldsymbol{i}} = \hat{\boldsymbol{T}}_{\boldsymbol{i}} \hat{\boldsymbol{D}}_{\boldsymbol{li}} \hat{\boldsymbol{T}}_{\boldsymbol{i}}^T$. $\hat{\boldsymbol{T}}_{\boldsymbol{i}}$ is the rotation matrix for the i-th lamina and it can be written as:

$$\hat{\boldsymbol{T}}_{\boldsymbol{i}} = \begin{bmatrix} m^2 & n^2 & mn \\ n^2 & m^2 & -mn \\ -2mn & 2mn & m^2 - n^2 \end{bmatrix} \tag{2.6}$$

where $m = \cos(\theta)$ and $n = \sin(\theta)$.

2.2. VAT structures

VAT laminates are characterized by fibers steered along curvilinear paths. Doing so, it is possible to increase the structural performances and to provide designers with more flexibility for trade-offs between structural properties.
The curvilinear path followed by the fibers can be described by different laws depending on the goals to be achieved. In [26, 34] the linear variation is adopted:

$$\theta(x') = \phi + (T_1 - T_0) \cdot \frac{|x'|}{d} + T_0. \tag{2.7}$$

where $x' = x\cos(\theta) + y\sin(\theta)$ is the direction along which fibers angle varies (the steering direction), d is a characteristic distance, T_1 is the fiber angle at $x' = d$, T_0 is the fiber angle at $x' = 0$ and ϕ is the rotation angle between the steering direction and X-axis (see Figure 2.4).

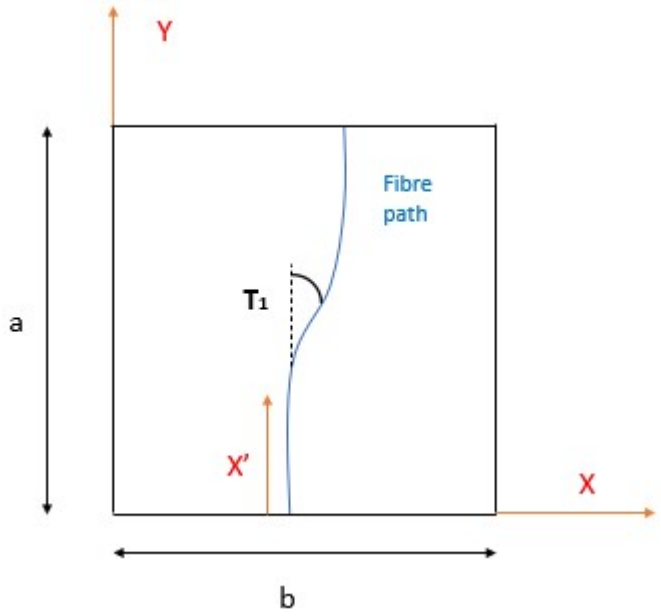


Figure 2.4: Fibers linear path for T_0=0° and generic T_1

Linear variation is the simplest type of path to be realized. In order to enlarge the design space, it is possible to adopt a non-linear path as the one in Figure 2.5. It can be defined, for example, using Lagrange polynomials, as it is done in [49].In the latter case the curvilinear path is described by:

$$\theta(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} T_{mn} \prod_{m\neq i} \frac{x - x_i}{x_m - x_i} \prod_{n\neq j} \frac{y - y_i}{y_n - y_i}. \tag{2.8}$$

where $(x_i; y_i)$ and $(x_j; y_j)$ are the x-y coordinates of the so-called reference points, while (M-1) and (N-1) are the number of these points along the x-direction and the y-direction, respectively. Moreover, the coefficients T_{mn} are the values of fiber angles at those points (defined by the designer).

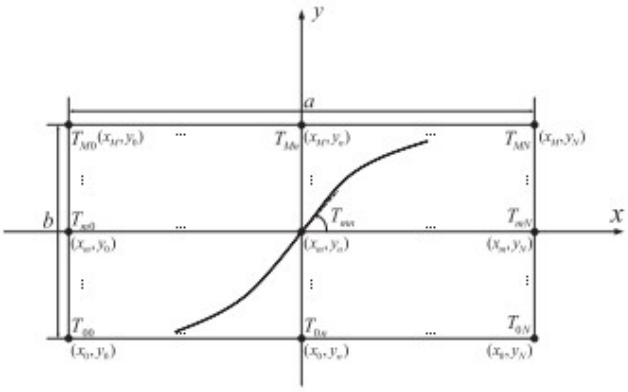
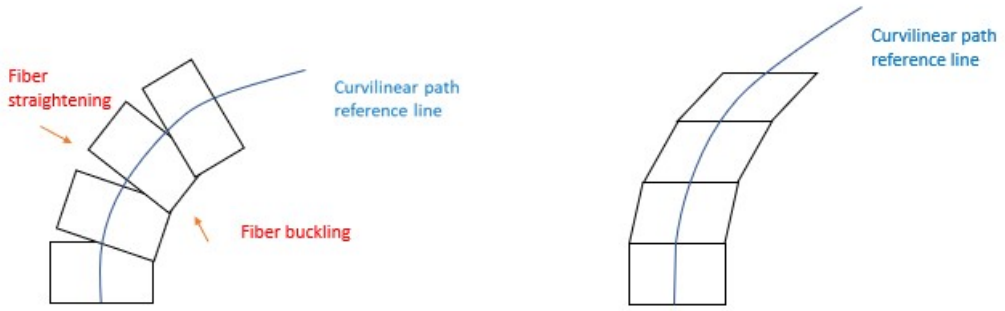


Figure 2.5: Non-linear variation of fibre orientation using Lagrange polynomials [47]

Concerning the in-plane elastic properties of the VAT plate, the steps necessary to define them are the same reported in the previous sections. The main difference is that, in this case, the stiffness matrix of each lamina $\hat{\boldsymbol{D}}_{\boldsymbol{i}}$ is function of the spatial coordinates (x,y), since the rotation matrix $\hat{\boldsymbol{T}}_{\boldsymbol{i}}$ is not constant as in the case of straight fibers. Therefore the plate stiffness matrix $\hat{\boldsymbol{D}}_{\boldsymbol{L}}$ is function of the in-plane spatial coordinate.

2.2.1. Manufacturing of VAT structures and related defects

Nowadays two techniques are mainly employed to manufacture variable-angle composites: Automatic Fibre Placement (AFP) and Continuous Tow Shearing (CTS). Both of them exploit a machine, endowed with several computer controlled axes, which places on the surface a band of split prepreg tape, called course (which is made of narrow fibre prepreg tapes called tows), with the help of a compaction roller and the heating to increase the tackiness. A ply is manufactured by placing multiple courses on the surface and by shifting them perpendicularly to the steering direction.
The main difference between the technique mentioned above is the way the fibres are steered: AFP exploits the in-plane bending deformation to achieve a curved tow path, whereas CTS exploits the shear deformation, as depicted in Figure 2.6.



(a) AFP steering technique (b) CTS steering technique

Figure 2.6: Steering techniques

The method employed to manufacture a VAT structure has drastic consequences on the type of defects exhibited by the manufactured laminate.
Regarding AFP, the main defects are fibers local buckling, presence of gaps and overlaps between adjacent courses and fibers out-of-plane waviness. The first one is due to the curvature of the tow path and can be minimized by keeping the radius of curvature below a certain value, named *'critical steering radius'* [3]. Gaps and overlaps are unavoidable. The out-of-plane waviness is a consequence of the latter when the structure undergoes the curing process and, so, it is unavoidable and can be mitigated only by limiting the presence of gaps and overlaps.

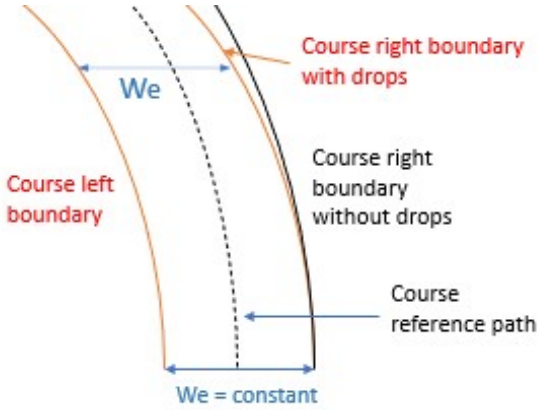


Figure 2.7: AFP course characteristics

Gaps and overlaps (see Figure 2.8) are unavoidable because, during the courses placement, the effective width w_e (see Figure 2.7) has to be kept constant in order to guarantee the complete coverage of the laminate surface. Consequently, some tows must be cut as soon as they reach the adjacent course or must overlap on the adjacent course. In the first

case, triangular area filled by resin arise, while in the second case a thickness variation is induced.

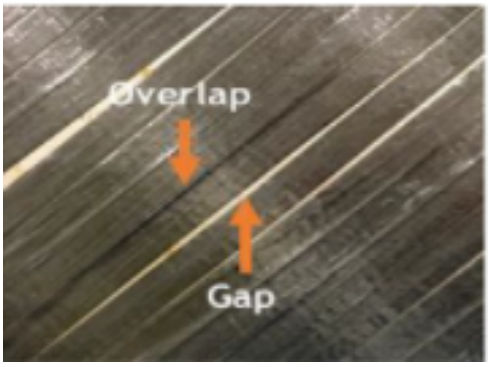


Figure 2.8: Gaps and overlaps [51]

The amount of gaps or overlaps can be decided by setting a certain coverage parameter. A coverage of 0% indicates that a tow is cut as soon as one of its edge reaches the boundary of the adjacent course (complete gaps strategy), whereas a 100% coverage means that the tow is cut only when its second edge crosses the adjacent course boundary (complete overlaps strategy), as depicted in Figure 2.9. Coverage values between these extremes represent the intermediate cases.

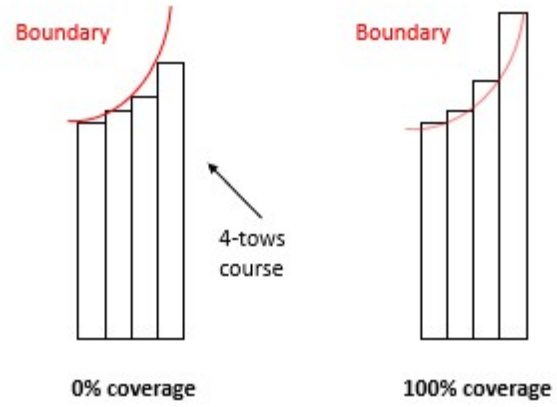


Figure 2.9: Strategies with different coverage parameters

The CTS technique was developed in order to minimize the amount of defects arising from the VAT structures manufacturing process, as explained by Kim et al. in [23]. This is possible because, as depicted in Figure 2.6, the in-plane shear deformation allows a perfect

tessellation of the tows. The only defect induced by the process is a thickness build up perpendicular to the fiber path. In particular, this variation of thickness is proportional to the fibers angle θ according to:

$$t = \frac{t_0}{cos(\theta)}, \tag{2.9}$$

where t_0 is the initial thickness of the lamina.

The defects mentioned above are strictly dependent on the manufacturing process and, so, their position can be traced back. Generally, the structure can be also affected by other unexpected defects, whose positions are unknown. Fibers misalignment, other gaps and overlaps, uncertainties on material and geometric properties belong to the this family of defects.

2.2.2. Numerical modeling of process-induced defects

Process-induced defects are responsible of mechanical properties alteration of VAT structures. Being able to quantify these variations is an important task to be assessed.
An experimental investigation conducted by Woigk et al. [46] was done to characterize the tensile and in-compression behaviour of defected specimens. The study shows that a specimen characterized by the simultaneous presence of gaps and overlaps exhibits a strength reduction of about 10% with respect to the pristine one. On the other hand, a numerical study conducted by Fayazbakhsh et al. [13] shows that in-plane stiffness and buckling load of a VAT plate can decrease of 16% and 12%, respectively, when adopting a complete gap strategy. Other studies were conducted, some of which combined experimental and numerical investigations [30].
From the results described above, it is clear that the effects of the defects are not negligible. It follows that it is necessary to quantify them, in order to be taken into account during the design phase. For this reason, numerical techniques were developed.

Concerning the gaps and overlaps, two modeling philosophies exist: one proposed by Blom et al. [4] and the other in [13]. Both methods have the same starting point, which is a MATLAB subroutine that allows to trace back the locations of the process-induced gaps and overlaps, as it is well explained in [4]. On the other hand, they differ a lot in other aspects.

In the first method, the main assumption is that each finite element is completely either in gap or regular composite material area, neglecting the thickness change due to overlaps (so it is recommended for a 0% coverage parameter). This approach requires the size of the elements to be small enough to capture the gap areas precisely (the size is governed by the width of a single tow), which means that the number of elements will drastically increase by enlarging the dimensions of the structure, while the computational efficiency will be reduced.

The main purposes of the second method proposed, called *'Defect Layer Method'*, are the reduction of the computational effort of the former technique and the introduction of a way to take into account the overlaps.
Differently from the previous method, an element doesn't have to be defined completely in a defected or regular composite area. It follows that each element, called defect element, is a regular composite element with modified properties with respect to the regular composite ones. In particular, the elastic properties are changed depending on the defect area percentage, which is the ratio between the defected area and the total area of the element.
For an element containing a gap, the elastic properties of the material are reduced, while if it contains an overlap only the thickness has to be changed. Elastic properties reduction and thickness variation are proportional to the value of defect area percentage.
In order to exploit this method, the choice of element size is driven by the requirement of limiting the difference between the fibers angles assigned to adjacent elements, which is less strict than the one of the previous method.
'Defect Layer method' showed to be more precise in capturing gaps and overlaps (see Figure 2.10) and, moreover, since the element size is bigger than the one necessary for the previous technique, it is recommended for modeling larger structure.

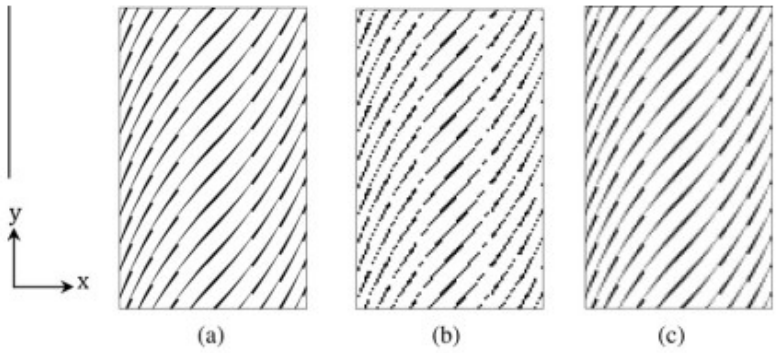


Figure 2.10: Geometry and location of gaps for a generic lamina obtained with: (a) true model,(b) Blom et al. approach (c) Fayazbakhsh et al. approach (Figure taken from [13])

Another study that is worth mentioning is the one conducted by McInnes et al.[31], which was focused on the modeling of thickness build-up consequence of CTS process (it can be considered valid also to model the thickness variation due to overlapping in AFP). In this paper, the main purpose was to understand if the adoption of continuum shell elements for modelling the thickness variation offers advantage with respect to the conventional shell elements in terms of results and computational effort. For this purpose, a comparison between the found buckling loads and natural frequencies of a plate and of a cylinder was performed.
The research concluded that using continuum shell elements allows for a better computational efficiency because of the capability to capture more precisely the thickness variation with a lower number of elements. Moreover, it was found that this advantage is more pronounced when dealing with lamina characterized by a grid-like thickness build-up (see Figure 2.11), which is obtained by when laminae with orthogonal steering directions are stacked together.

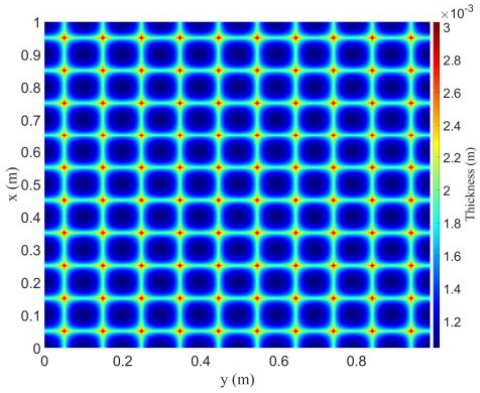


Figure 2.11: Grid-like thickness build-up considering $\phi_1 = 0$ and $\phi_2 = 90$ [31]

Regarding the out-of-plane waviness, in [32] it was illustrated the generation of a 3D finite element model able to reproduce fibers out-of-plane waviness, starting from real specimens optical images and experimental tests. The numerical analyses showed that a pronounced waviness can be very problematic because it could be the reason of high interlaminar strains, which is one of the main causes of early failure for composite laminates.

To summarize, the defects are source of uncertainty about the structural response and, so, the modelling techniques presented earlier can be employed to quantify this uncertainty through the so-called sensitivity analyses, which are treated in the next chapter.

3 | Sensitivity analyses

Defects are source of uncertainty for the mechanical performances of the structures. An effective to quantify their effects are the sensitivity analyses (SA). Usually, these analyses are carried out using the Stochastic Finite Element method (SFEM), which is time consuming. In order to speed up the process the SFEM can be used together with the so-called *'surrogate models'* in the Spectral Stochastic Finite Element Method (SSFEM). In this chapter, the SFEM and the SSFEM are illustrated, highlighting their differences. Moreover, a deepening on the Polynomial Chaos Expansion (PCE) and Karhunen-Loeve Expansion (KLE) surrogate models is provided.

3.1. Sensitivity analysis using Stochastic Finite Element Method (SFEM)

Once the uncertain inputs of the structural model are identified, the sensitivity analysis consists in a large number of structural analyses on the same structure. For each analysis different values are assigned to the uncertain inputs. Since the solution of structural problems usually requires to deal with analytically unsolvable PDEs, each analysis is performed using numerical methods, such as finite elements.

Considering stochastic inputs in the form $\mathbf{V}(\mathbf{x}, \alpha)$, where α represent their random nature, the SFEM requires a certain number of samples α_1,...,α_m, which are obtained through Monte Carlo simulation.
Once a sample of stochastic inputs $\boldsymbol{V}(\boldsymbol{x}, \alpha_i)$ is obtained, the problem becomes a deterministic one. So, it can be solved as a canonical FE problem in order to obtain the desired outputs $\boldsymbol{U}(\boldsymbol{x}, \alpha_i)$. When it can be assumed that the space of outputs possible solutions is correctly sampled, it is possible to calculate the statistical second-order moments of the quantities of interest.
The Monte Carlo method is typically used for sampling stochastic quantities because of its simplicity. Alternative strategies, however, are also possible. An example is the Covari-

ance matrix decomposition [9]. In particular, the paper shows how to produce simulations of the random variable of interest from the LU decomposition of the covariance matrix, which is built from a grid $N \times M$ of points on the structure.
An application of this method can be found in the work of Pagani et al. [33], where samples of fibers misalignment distribution, adopting a squared exponential correlation function, were produced.

The SFEM is a consolidated method to make sensitivity analyses, but the amount of numerical analyses necessary to obtain reliable results is very large, typically of the order of 10^4. In order to reduce the computational time, alternative strategies can be employed. The usage of surrogate models represents a viable approach to speed up sensitivity analyses.

3.2. Surrogate models

In order to decrease the amount of computational effort many activities were performed to prove the reliability of different types of *surrogate models*.
When surrogate models are adopted, the structural output of interest is expressed through an expansion made of functions dependent on variables that parameterize the uncertain inputs. Then, each function is pre-multiplied by an unknown coefficient, which has to be tuned to build the deterministic expression of the series.
The main advantage of this models is that only a reduced amount of numerical analyses must be performed. Their number must be large enough to find the correct values of the coefficients. As a matter of fact, once they are tuned, it is only necessary to substitute any other inputs sample inside the approximating expansion to calculate the correspondent quantity of interest. It results that the determination of output statistical moments is much faster.

An interesting review of the most common surrogate models is provided in [44]. Among the listed strategies in [44], one of the most used is the *Polynomial Chaos Expansion* (PCE) introduced in [45]. Usually, it is employed together with the *Karhunen-Loeve Expansion* (KLE) in the Spectral Stochastic Finite Element method (SSFEM) to run sensitivity analysis [14].

3.2.1. Polynomial Chaos Expansion (PCE)

Polynomial Chaos Expansion was introduced in [45] aiming to represent well-behaved random variables, as the second-order ones. The representation is done by means of an expansion made of orthogonal polynomials dependent on a set of N_γ independent random variables $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_i, ..., \gamma_{N_\gamma})$.
From a mathematical point of view, if $Q(\alpha)$ is the uncertain second order variable, where α stands for its randomness, its PCE is $c(\alpha)$:

$$Q(\alpha) \approx c(\alpha) = a_0\Gamma_0 + \sum_{i=1}^{N_\gamma} a_{1i}\Gamma_1(\gamma_i(\alpha)) + \sum_{i=1}^{N_\gamma}\sum_{j=1}^{i} a_{2ij}\Gamma_2(\gamma_i(\alpha), \gamma_j(\alpha)) + ... \tag{3.1}$$

In Eq. (3.1), Γ_i belongs to the set of polynomials of order i-th of the orthogonal basis chosen for the expansion, while $\boldsymbol{a} = (a_0, a_{11}, ..., a_{1i}, ..., a_{1N_\gamma}, a_{211}, ..., a_{2ij}, ..., a_{2N_\gamma N_\gamma}, ...)$ is the vector of coefficients to be tuned and their first subscript stands for the order of the polynomials pre-multiplied.
If the second order variable $Q(\alpha)$ is Gaussian, the Hermite polynomials are the orthogonal basis for the expansion and each random variable γ_i must be a centered normalized Gaussian variable. Moreover, the number of Gaussian variables is limited to N_γ, but, theoretically, they should be infinite. As consequence, the number of Hermite polynomials necessary to build the expansion would be infinite, too. For computational reasons the expansion is truncated. In particular, assumed that p is the maximum polynomial order and that N_γ is the number of Gaussian variables, it turns out that the number of terms in the expansion is equal to P :

$$P = \frac{p + N_\gamma}{p!N_\gamma!}, \tag{3.2}$$

The truncated series is proved to converge in the L_2 sense for any stochastic process with finite second-order moments [6]. Moreover, the convergence rate is optimal if the expansion approximates a Gaussian process, but it can be slow for the non-Gaussian ones. For this reason, a summary of the polynomials to be used for different probability distributions is provided in Table 3.1 [48].

$Q(\alpha)$ probability distribution	Polynomial Basis for PCE
Gaussian	Hermite
β	Jacobi
γ	Laguerre
Uniform	Legendre

Table 3.1: Examples of polynomial basis to use for Gaussian and non-Gaussian processes

As discussed earlier, the PCE can be used as a surrogate model for the sensitivity analysis of structures. In [7, 29] the PCE was used to approximate a generic stochastic output u dependent on random inputs, which were parameterized by means of the set of independent random variables $\boldsymbol{\gamma}$. In both papers, once the expansion in Eq. (3.1) was written using the best polynomial basis from Table 3.1, the values of each coefficient in $\boldsymbol{a}$ were calculated through the for the least-square approximation.
In particular, considered N sets of N_γ random variables and the correspondent structural output u_j for j=1,...,N (i.e. obtained by numerical analysis), the vector of coefficients $\boldsymbol{a}$ can be found solving the following least-square problem:

$$\boldsymbol{a} = arg \quad \min_{a} \sum_{i=1}^{N} (u_i - c_i(\alpha))^2. \tag{3.3}$$

where $c_i(\alpha)$ is the value of the PCE once a set of random variables is substituted into Eq. (3.1).
Once the coefficients are tuned, the second-order moments of the structural output, the mean value and variance, can be calculated as follows:

$$\begin{cases} \mu_u = a_0, & \text{(3.4a)} \\ \sigma^2 = \sum_{r=1}^{P} a_r^2. & \text{(3.4b)} \end{cases}$$

3.2.2. Karhunen-Loeve expansion (KLE)

The Karhunen-Loeve expansion is used to represent stochastic quantities characterized by a certain correlation function and variance. This is done through the spectral decomposition of its covariance function.
In particular, given a stochastic field $K(\boldsymbol{x}, \alpha)$ with covariance function C$(\boldsymbol{x}_1, \boldsymbol{x}_2)$ (where

$\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are the vectors of spatial coordinates), it is possible to express it as the sum of its mean value and an infinite series expansion: the Kharunen-Loeve expansion (KLE). It follows that $K(\boldsymbol{x}, \alpha)$ can be written as:

$$K(\boldsymbol{x}, \alpha) = \mu_k(\boldsymbol{x}) + \sum_{i=1}^{\infty} \xi_i(\alpha)\sqrt{\lambda_i}\phi_i(\boldsymbol{x}). \tag{3.5}$$

where $\boldsymbol{x}$ is the vector of spatial coordinates, $\xi_i(\alpha)$ is an uncorrelated random variable and (λ_i,$\phi_i(\boldsymbol{x})$) are, respectively, the i-th eigenvalue and i-th eigenfunction of the covariance function.
The eigenfunctions are orthogonal to each other and they belong to a complete basis in $L^2(\Omega)$ space of square integrable function in Ω. The uncorrelated variables are centered normalized Gaussian variables if the stochastic field is Gaussian, whereas for non-Gaussian process they are generally non-Gaussian and their distribution must be found through iterative process [36].

The eigenvalues and eigenfunctions have to be found by solving the homogeneous Fredholm integral of the second kind:

$$\int C(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi_i(\boldsymbol{x}_2)d\boldsymbol{x}_2 = \lambda_i\phi_i(\boldsymbol{x}_1). \tag{3.6}$$

The solution of Eq. (3.6) is the most challenging thing to do in order to exploit the KLE. As a matter of fact, the analytical solution exists only for few types of correlation functions (i.e. the exponential one), as stated in [14].
For other types of correlation function it is necessary to use numerical methods.

As seen for the PCE, the KLE theoretically is made of infinite terms, but the series is truncated for computational reasons. However, considered a certain number of terms, the KLE is optimal with respect to the global mean square error among every series expansion method.
To verify if the number of terms employed is acceptable, comparing the real covariance function and the one approximated by the expansion is a good strategy. In particular, from the spectral decomposition of the covariance function, its approximation is equal to $\bar{C}(\boldsymbol{x}_1, \boldsymbol{x}_2)$:

$$\bar{C}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{i=1}^{N_T} \lambda_i \phi_i(\boldsymbol{x}_1)\phi_i(\boldsymbol{x}_2). \tag{3.7}$$

where N_T is the number of terms in the expansion.
For example, in Figure 3.1 is depicted the comparison between the theoretical 1D exponential covariance and the approximated one for a 10 terms KLE.

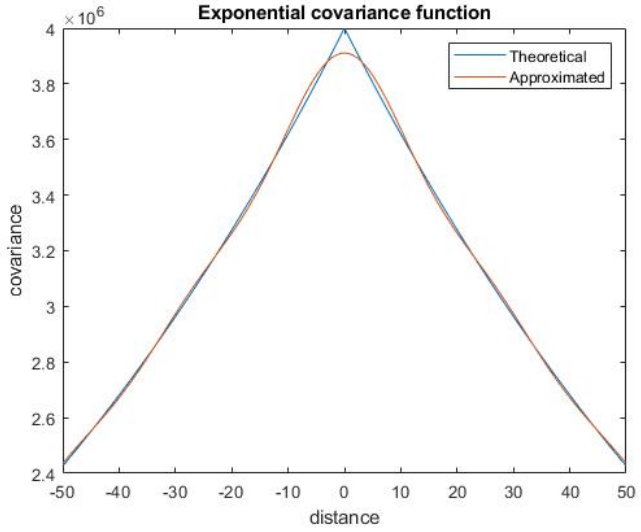


Figure 3.1: Covariance functions comparison

3.2.3. Spectral Stochastic Finite Element Method (SSFEM)

Usually, the solution of a structural problem implies the solution of Partial Differential Equations (PDEs), which could be unsolvable analytically.
In this case, the problems are solved numerically exploiting the Finite Element Method (FEM). When everything is known about the problem, it is called *'deterministic'* finite element method and the outputs are deterministic. Sometimes, some variables (i.e. the material Young Modulus) could be uncertain and they need to be considered as stochastic variables, not as deterministic ones. As a consequence, the system under investigation is governed by stochastic PDEs and, so, its outputs are random. The numerical solution of these problems requires the *'stochastic'* finite element method (SFEM), previously illustrated in Section 3.1.

A variant of the SFEM was proposed by Ghanem and Spanos in [14]: the Spectral Stochas-

tic Finite Element Method (SSFEM).
In this method, the uncertain inputs are expressed by means of Karhunen-Loeve expansions, while the outputs by means of PCEs. In particular, the method can be summarized as follows (see [42] for more details):

1. Building the KLE of each stochastic input using a set of N_γ random variables (Section 3.2.2);
2. Building the PCEs of the stochastic outputs using each random variable employed to build the KLEs (Section 3.2.1);
3. Sampling of the inputs (by sampling the random variables $\xi_i(\alpha)$ using Latin Hypercube sampling, for example) in order to assign the properties to the finite element model and to perform the FE analysis for collecting the outputs samples;
4. Tuning of the PCEs coefficients (Section 3.2.1).

The main advantage of the SSFEM is that it is not necessary to perform as many analyses as required in the SFEM. As a matter of fact, the number of analyses must be large enough to tune the PCEs coefficient. It follows that the computational effort and the time required to obtain reliable statistical second-order moment of the outputs are much lower.
Moreover, at the end of the process, there are available analytical expression of the uncertain outputs, which are function of the variables that parameterize the stochastic inputs.

Since its formulation, the SSFEM was employed for the solution of many stochastic structural problems, as in [20, 39].
In [39] the method was employed for the sensitivity of a VAT laminate characterised by uncertain fiber misalignment and uncertain volumetric fraction as inputs. For each ply, the authors followed these steps: they created two KLE to describe the fiber misalignment and the volumetric fraction, they sampled these stochastic fields by generating the value of the random variables $\xi_i(\alpha)$ by means of Latin Hypercube Sampling and, then, they assigned these variables to the finite elements to perform the buckling analysis. A surrogate model was built for each buckling load by means of the PCE, whose coefficients were tuned exploiting the result of the numerical analysis. In the end, statistical second-order moments were calculated.
An interesting thing about this paper is the comparison between the SSFEM and SFEM in terms of results and computational effort. The authors showed that the computation

of the statistical moments through SFEM required 72 hours of computational time, while the SSFEM only 21 hours (most of it employed for the PCE construction).

In [20] 1D and 2D problems were considered.
The 1D problem was about a beam of length $2a$ and loaded in bending by a concentrated force applied at its tip. The random input of the problem was the Young modulus, while the random output of interest was the tip displacement. In particular, the Young modulus was considered as a Gaussian process and it was represented through a Karhunen-Loeve expansion built considering an exponential covariance function, a certain value of variance σ^2 and correlation length c. In this case, the solution of Eq. (3.6) is analytical:

$$\begin{aligned} \lambda_i &= \frac{2\sigma^2 c}{\omega_i^2 + c^2} \qquad \text{for odd i,} \\ \lambda_i &= \frac{2\sigma^2 c}{\omega_{i*} i^2 + c^2} \quad \text{for even i.} \end{aligned} \tag{3.8}$$

$$\begin{aligned} \phi_i(x) &= \frac{cos(\omega_i x)}{\sqrt{a + \frac{sin(2\omega_i a}{2\omega_i}}} \qquad \text{for odd i,} \\ \phi_i(x) &= \frac{sin(\omega_{i*} x)}{\sqrt{a + \frac{sin(2\omega_{i*} a}{2\omega_{i*}}}} \quad \text{for even i.} \end{aligned} \tag{3.9}$$

where ω_i and ω_{i*} are the solution of the following equations:

$$\begin{aligned} c - \omega tan(\omega x) = 0, \\ \omega_* + c tanh(\omega_* x) = 0. \end{aligned} \tag{3.10}$$

The tip displacement, instead, was represented by means of a PCE of second order using the Hermite polynomials. In particular, if the KLE is built by using only two random variables $\xi_i(\alpha)$, the Hermite polynomials until the second order are:

$$(\xi_1, \xi_2, \xi_1^2 - 1, \xi_2^2 - 1, \xi_1 \xi_2). \tag{3.11}$$

For the 2D case, the stochastic input was the Young modulus of a plate (still considered as a Gaussian process), while the random output was the vertical displacement in a generic point. The way the problem was solved was similar to the 1D case, since the KLE was used to express the Young modulus and the PCE for the vertical displacement, but it is worth mentioning how to obtain the KLE.
For an exponential covariance function $C_e(x_1, x_2, y_1, y_2)$:

$$C_e(x_1, x_2, y_1, y_2) = \sigma^2 e^{(-|x_1-x_2|/c_x)} e^{(-|y_1-y_2|/c_y)}, \tag{3.12}$$

it is possible to assume that the Fredholm equation of the second kind can be solved separately for the x and y directions.
So, the eigenvalues and eigenfunctions to be used in the KLE are:

$$\lambda_n = \lambda_i^{(x)} \lambda_j^{(y)} \quad \text{for i=1,...,} N_\xi \text{ and for j=1,...,i.} \tag{3.13}$$

$$\phi_n(x, y) = \phi_i(x)\phi_j(y) \quad \text{for i=1,...,} N_\xi \text{ and for j=1,...,i.} \tag{3.14}$$

where $\lambda^{(x)}, \lambda^{(y)}, \phi(x)$ and $\phi(y)$ can be found using Eq. (3.8) and Eq. (3.9), respectively. Moreover, N_ξ depends on the number of terms set for the KLE.

4 | Neural networks

'Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment' [12]. Neural networks fall in this category of techniques.
The latter, in particular, are inspired to the early models of sensory processing by the brain and they have the goal to solve many types of problems, like classification, regression, control, modelling, forecasting and many others. Moreover, they are becoming more and more important for the solution of engineering problems, like the detection of structural defects. For these reasons, they are an important tool for the development of the thesis.
In this chapter, the mathematical background regarding the implementation and the training process of two variant of neural networks, Artificial Neural Network (ANN) and Physics Informed Neural Network (PINN), is illustrated.

4.1. Artificial Neural Networks (ANN)

Artificial neural networks (ANNs) are mathematical models based on many units named *neurons*, which are organized in layers. In general, each neuron is connected to the others by means of a coefficient, the *weight* w, while each neuron in the network is associated to a coefficient called *bias* b.
In particular, the most common type of ANN employed is the one named *feed-forward*, in which each neuron is only connected to the others belonging to the subsequent layers. This means the information flows in one direction only, from the input layer, through the hidden layers, until the output layer (see Figure 4.1).

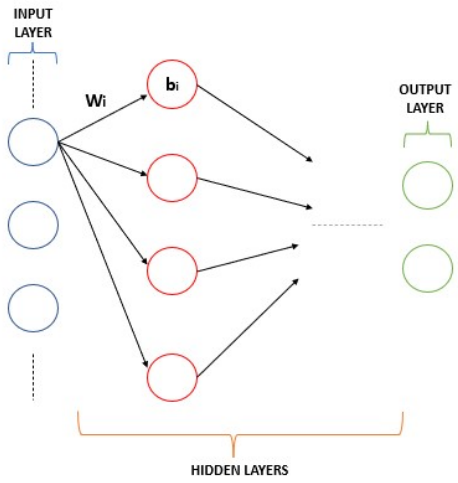


Figure 4.1: Feed-forward ANN architecture with one hidden layer

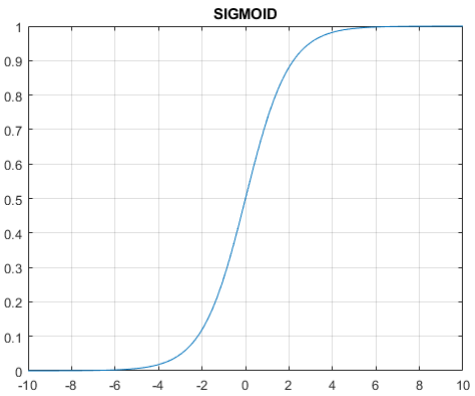
To solve a problem, the ANN needs set of N_S labeled inputs $\boldsymbol{X_L}$ and the corresponding known set labeled outputs $\boldsymbol{U_L}$. The labeled input vector enters the neural network through the input layer and is manipulated in order to obtain the network outputs vector $\boldsymbol{O}$.
In particular, for a neural network with N_H hidden layers, the outputs vector is found as:

$$\boldsymbol{O} = \hat{\boldsymbol{W}}_{\boldsymbol{u}} \cdot \boldsymbol{h}^{(N_H)}, \tag{4.1}$$

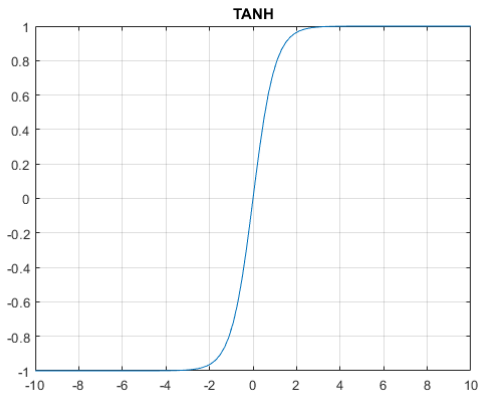
where $\hat{\boldsymbol{W}}_u$ is $N \times M$ matrix (N=number of output layer neurons, M=number of neurons in the last hidden layer) which contains the output weights that connects the N_H-th hidden layer neurons to the output layer (ordered row by row) and $\boldsymbol{h}^{(N_H)}$ a vector $M \times 1$ which is the output of the N_H-th hidden layer.
Assuming that $\boldsymbol{h}^{(0)} = \boldsymbol{X_{Li}}$ (where $\boldsymbol{X_{Li}}$ is a generic sample from the set $\boldsymbol{X_L}$), the generic output $\boldsymbol{h}^{(i)}$ of the i-th hidden layer is:

$$\begin{aligned} \boldsymbol{h}^{(i)} &= a^{(i)}(\hat{\boldsymbol{W}}_{INP}\boldsymbol{h}^{(i-1)} + \boldsymbol{b}^{(i)}) \qquad \text{for i=1,} \\ \boldsymbol{h}^{(i)} &= a^{(i)}(\hat{\boldsymbol{W}}_i\boldsymbol{h}^{(i-1)} + \boldsymbol{b}^{(i)}) \qquad \text{for i=2,...,}N_H. \end{aligned} \tag{4.2}$$

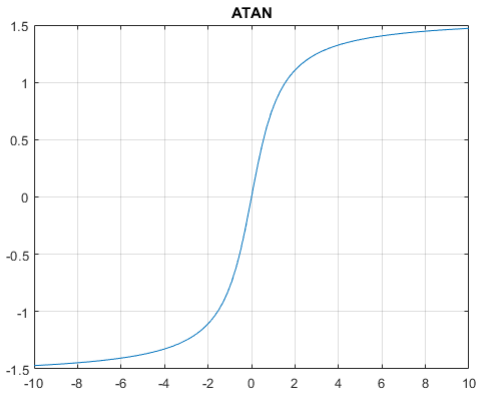
where $\hat{\boldsymbol{W}}_i$ is $M \times M$ matrix which contains the i-th hidden layer weights row by row, $\hat{\boldsymbol{W}}_{INP}$ an $M \times I$ matrix containing the input weights and $\mathbf{b}^{(i)}$ is the vector of biases of the i-th hidden layer and $a^{(i)}(\cdot)$ is the activation function, which operates a non-linear transformation for mapping the neuron outputs in a dimensionless domain. In particular, *Tanh, sigmoid* and *atan* are the most used activation functions (see Figure 4.2).



(a) SIGMOID



(b) TANH



(c) ATAN

Figure 4.2: Activation functions

An important feature of the neural network is the definition of a loss function L, which quantifies the error between the known labeled outputs vector $\boldsymbol{U_L}$ and the neural networks outputs vector $\boldsymbol{O}$:

$$L = \sum_{i=1}^{N_S} \frac{|\boldsymbol{U_L}_i - \boldsymbol{O}_i|^2}{2N_S}, \tag{4.3}$$

The objective is minimizing the value of the loss function by finding the optimal values of the *internal parameters* vector $\boldsymbol{\theta}$, which are the biases and the weights. This can be done by *'training'* the network using optimization algorithms that will be shown.

4.2. Physics-Informed Neural Network (PINN)

Neural networks are widely used to solve problems in solid mechanics, like the solution of Partial Differential Equations (PDEs) expressing the equilibrium conditions.
The typical ANNs can solve forward problems. Starting from available sets of data, they learn the behaviour of the system through a the training process. Since only labelled data are used for the training process, this type of neural networks are named *black-box ANNs*. Sometimes, the data necessary to train the *black-box ANNs* is limited and, so, it is necessary to enrich the problem statement with other known information, like the physical laws and boundary conditions that govern the behaviour of the systems under investigation. In this case the neural network works as a *grey-box* and is named Physics-Informed Neural Network (PINN).
The main difference with respect to the ANN is that the loss function to be minimized takes into account the contributions regarding the equilibrium equations and boundary conditions residuals evaluated at a set of N_{CD} and N_{CB} *'collocation points'*, respectively. In this case, it is possible to distinguish three contributions in the loss function L: the data-driven contribution L_{DD}, the physics-driven contribution L_{CD} regarding the equilibrium equations residuals and the physics-driven contribution L_{CB} regarding the boundary conditions residuals:

$$\begin{aligned} L_{DD} &= \sum_{i=1}^{N_S} \frac{|U_i - O_i|^2}{2N_S}, \\ L_{CD} &= \sum_{i=1}^{N_{CD}} \frac{|f_i|^2}{2N_{CD}}, \\ L_{CB} &= \sum_{i=1}^{N_{CB}} \frac{|b_i|^2}{2N_{CB}}, \end{aligned} \tag{4.4}$$

where f_i and b_i are the equilibrium equations and boundary condition residuals evaluated at the i-th collocation point of the correspondent set.
The PINNs can be enriched further in order to help more the convergence to the solution, as it is done in [50]. This paper started from the concept that since the residuals of the equilibrium equations must be zero, also the corresponding gradients must be zero (and so the second derivative, third derivative, ecc...)and, so, this information was used in the training process of PINNs.
The study showed that adding the gradient residuals was useful to achieve better results with respect to the ones obtained without including these further information.
This variant of PINN goes under the name gPINN.

The popularity of PINNs is also due to the fact that, from the implementation point of view, there is no difference in solving forward or inverse problems. In particular, inverse problems are those that, besides simulating system behaviour, finds uncertain parameters regarding the system underlying model. This happens when the structural problems are not well-defined, but are characterized by unknown parameters.
In order to find these unknown parameters, they are added to the internal parameters vector $\boldsymbol{\theta}$ of the network. Doing so, they can be tuned together with the output weights during the training process.

This ability in solving inverse problems is exploited in the method developed in this thesis. As a matter of fact, defects distributions will be written as function of structural parameters. Then, these parameters will be added to the internal parameters vector $\boldsymbol{\theta}$ in order to be tuned during the training process. Once completed the training process, the tuned parameters can be used to reconstruct the spatial distribution of defects.

4.3. Training process

Training is the process through which the network learns the internal parameters.
The most common ways to train a network are the Back Propagation (BP) algorithms and the Extreme Learning Machine (ELM) algorithm.
The mathematical backgrounds behind these algorithms is presented in this section.

Back Propagation (BP)

Back propagation is a class of optimization algorithms that exploits the computation of the loss function gradient with respect to the internal parameters in order to update them. To this family of algorithms belong the gradient descent algorithm (GB, the most common) [38] and the Adam optimization [24] algorithm. Other examples can be found in [5].
In order to explain the working mechanism of GB optimization, the following nomenclature is introduced:

- *batch-size*: number of samples from a dataset;
- *epochs*: one round of training on a certain dataset;
- *shuffle* : recombination of a dataset;

Three variants of gradient descent algorithms exist:

1. **batch gradient descent**. In this case the entire dataset is used to build the loss function L.
   In paricular, the gradient $\boldsymbol{J}$ of the loss function with respect to the internal parameters is computed and evaluated in correspondence of the current $\boldsymbol{\theta}$. During this step, *automatic differentiation* algorithms are employed to compute the derivatives [2].
   Then, the vector of internal parameters is updated as it follows:

$$\boldsymbol{\theta_i} = \boldsymbol{\theta_{i-1}} - \eta \mathbf{J}(\boldsymbol{\theta_{i-1}}) \quad \text{for i=1,...,} N_E. \tag{4.5}$$

   where η is a tunable parameter, named *'learning rate'*, fixed in order to limit the internal parameters variation for an update, N_E is the number of epochs and $\boldsymbol{\theta_0}$ is the initial guess for the internal parameters vector. The maximum number of epochs performed depends on the arrest criterion adopted. Usually, a maximum number for the iteration is specified; alternatively, the process ends when the norm of the difference between the internal parameters vector at two consecutive iterations is smaller than the tolerance. Moreover, the dataset is shuffled before each epoch.

   This algorithm guarantees convergence to global minimum for convex error surfaces. On the contrary, there is no guarantee to converge to the global minimum for non-convex surfaces. In this case, the algorithm may find a local minimum.

2. **stochastic gradient descent** Differently from the previous variant, this one exploits one dataset sample, one for each iteration, to build the loss function L.
   It follows that the internal parameters vector is update as follows:

$$\boldsymbol{\theta_i} = \boldsymbol{\theta_{i-1}} - \eta \mathbf{J}(\boldsymbol{\theta_{i-1}}, Sample_i) \quad \text{for i=1,...,} (N_S \times N_E). \tag{4.6}$$

   where N_S is the number of samples in the dataset.
   Also in this case, the training process ends when one of the arrest criteria explained previously is satisfied.

   Regarding the performance, this variant is faster than the previous one, while showing the same convergence features.

3. **mini-batch gradient descent**
   In this case the dataset is divided in multiple mini-batches of N_B samples each one. For each iteration, a different mini-batch is used to build the loss function, compute its gradient $\boldsymbol{J}$ with respect to the internal parameters and update the internal parameters vector $\boldsymbol{\theta}$.
   In particular, the internal parameters vector is updated as follows:

$$\boldsymbol{\theta_i} = \boldsymbol{\theta_{i-1}} - \eta \mathbf{J}(\boldsymbol{\theta_{i-1}}, batch_i) \quad \text{for i=1,...,}(N_B \times N_E). \tag{4.7}$$

   Also in this case, the considerations made about the arrest criteria are valid.

   Because of its efficiency, this variant is usually the one employed for the training process [38].

4.3.1. Extreme Learning Machine (ELM) for forward problems

Learning speed of neural networks relying on back propagation algorithms is very slow because many parameters have to be tuned iteratively at each update. It follows that convergence can be reached after several iterations and this aspect is one of the drawbacks regarding the usage of neural networks.

A new training algorithm was proposed in [19] with the aim of increasing the learning speed: the Extreme Learning Machine (ELM).
This type of algorithm can be used for Single-hidden Layer Feed-forward (SLFN) networks, which are networks characterized by only three layers of neurons: the input layer, the output layer and a single hidden layer.
Previous studies [18] shown that SLFN with $\bar{N}$ hidden neurons can exactly learn $\bar{N}$ distinct observation by tuning the internal parameters (input and output weight, hidden layer biases). In [19] the authors proved that the same result can be obtained by assigning random value to input weights and hidden layer biases if the activation functions for the hidden layer are infinitely differentiable. It turns out that for SLFN employing ELM the internal parameters of the network are only the output weights.

The mathematical formulation of ELM is briefly outlined next. Considering a SLFN with $\bar{N}$ hidden neurons and a dataset of N_S distinct labeled observation ($\boldsymbol{X_{Lj}}$,$\boldsymbol{O_{Lj}}$), where

$\boldsymbol{X}_j$ is an $M \times 1$ vector and $\boldsymbol{O}_j$ is a $N \times 1$ vector, the network output $\boldsymbol{U}_j$ can be found following these two steps:

$$\begin{aligned} &1.h_{ij} = g(\hat{\boldsymbol{W}}_{INP,i}\boldsymbol{X}_{\boldsymbol{L}j} + b_i) \quad \text{for i=1,...,}\bar{N} \quad \text{for j=1,...,}N_S, \\ &2.\boldsymbol{U}_j = \sum_{i=1}^{\bar{N}} \hat{\boldsymbol{W}}_{OUT,i} h_{ij} \qquad\qquad\qquad\qquad \text{for j=1,...,}N_S. \end{aligned} \tag{4.8}$$

where h_{ij} is the output of the i-th hidden neuron due to the j-th input sample, $\mathbf{W}_{INP,i}$ is a vector $1 \times M$ of input weights connecting the inputs with the i-th hidden neuron, b_i is the bias of the i-th hidden neuron, g is the hidden layer activation function and $\mathbf{W}_{OUT,i}$ is the vector $N \times 1$ of output weights that connect the i-th hidden neuron to the outputs. From *theorem 1.1* reported in [19], it follows that once it is selected an infinite differentiable activation function for the hidden layer and the input weights and hidden layer biases are chosen randomly, the output weights can be tuned in order to satisfy $\boldsymbol{O}_{\boldsymbol{L}j} = \boldsymbol{U}_j$ for j=1,...,N if $\bar{N} = N$. In particular, the following linear system must be solved:

$$\hat{\boldsymbol{H}}\hat{\boldsymbol{W}}_{OUT} = \hat{\boldsymbol{O}}. \tag{4.9}$$

where $\hat{\boldsymbol{H}}$ is a constant matrix $N_S \times \bar{N}$, named *'Hidden layer output matrix'*, with the following form:

$$\hat{\boldsymbol{H}} = \begin{bmatrix} h_{11} & \dots & h_{\bar{N}1} \\ \vdots & \ddots & \vdots \\ h_{1N_S} & \dots & h_{\bar{N}N_S} \end{bmatrix} \tag{4.10}$$

Then, $\hat{\boldsymbol{W}}_{OUT}$ is a matrix $\bar{N} \times N$, in which the i-th row contains the output weights that connect the i-th hidden neuron to the outputs and $\hat{\boldsymbol{O}}$ is the known outputs matrix of $N_S \times N$ dimensions.
It is straightforward to understand that if $\bar{N} = N$ the $\hat{\boldsymbol{H}}$ matrix is square and, so, it can be inverted to find the value of $\hat{\boldsymbol{W}}_{OUT}$ that solve exactly Eq. (4.9).

If N is not equal to $\bar{N}$ the $\hat{\boldsymbol{H}}$ matrix cannot be inverted. In this case, according to *theorem 1.2* in [19], an unique combination of output weight that minimize the training error $||\hat{\boldsymbol{H}}\hat{\boldsymbol{W}}_{OUT} - \hat{\boldsymbol{O}}||$ exists. This combination is the minimum least-square solution and it is found by solving the following linear system:

$$\hat{\boldsymbol{W}}_{OUT} = \hat{\boldsymbol{H}}^*\hat{\boldsymbol{O}}. \tag{4.11}$$

where $\hat{\boldsymbol{H}}^*$ is the *'Moore-Penrose generalized inverse'*[1] of the $\hat{\boldsymbol{H}}$ matrix.

4.3.2. 'Extreme Learning Machine for inverse problems

In [11] Dwivedi et al. introduced a new variant of neural network: the Physics-informed Extreme Learning Machine neural network (PiELM). This variant exploits both the advantages of the ELM and the PINN, which are the higher learning speed and the need for less labeled data, respectively. In this paper, it was employed to solve different kinds of forward time-dependant problems and it was proved that there is no loss in the accuracy of the solution with respect to the canonical numerical techniques.
However, the application of this new technique is not restricted to the forward problems case only, but it can be adopted for inverse problems, too.

Differently from what it is shown in the previous section, the solution is not straightforward as for Eq. (4.9) or Eq (4.11).
As a matter of fact, the equilibrium equations of the system are non-linear dependently on the internal parameters of the network. This means that the $\hat{\boldsymbol{H}}$ matrix is not constant as in case of forward problems, but is a function of the internal parameters. It results that the solution of Eq. (4.9) can't be found by simply inverting the $\hat{\boldsymbol{H}}$, but it must be found through an iterative process, whose mathematical formulation is going to be illustrated.
Considering a neural network with M neurons in the input layer, N neurons in the output layer, $\bar{N}$ neurons in the hidden layer and N_δ uncertain parameters in the equilibrium equations, the internal parameters vector $\boldsymbol{\theta}$ is a vector of dimensions $((N \times \bar{N}) + N_\delta) \times 1$:

$$\boldsymbol{\theta} = [\boldsymbol{W}_{out1}, ..., \boldsymbol{W}_{outN}, \boldsymbol{\delta}]^T. \tag{4.12}$$

where $\mathbf{W}_{outi}$ is a vector $1 \times \bar{N}$ containing the weights connecting the last hidden layer the the i-th output neuron of the output layer, while $\boldsymbol{\delta}$ is the vector of unknown parameters.
As explained before, also in this case the solution of the neural network is the minimum least-square solution of a system like the following:

$$\hat{\boldsymbol{L}}\boldsymbol{\theta} = \boldsymbol{F} \tag{4.13}$$

where $\hat{\boldsymbol{L}}$ is a matrix $(N \times N_S + N_C) \times (N \times \bar{N} + N_\delta)$, in which N_C is the number of collocation points in which the equilibrium equations and/or boundary conditions are evaluated, and $\boldsymbol{F}$ is the vector of known outputs.

The $\hat{\boldsymbol{L}}$ matrix can be built has the composition of two parts, the data-driven part and the physics-driven one.
The data-driven matrix is constant and is equal to $\hat{\boldsymbol{L}_{\boldsymbol{DD}}}$:

$$\hat{\boldsymbol{L}}_{\boldsymbol{DD}} = \begin{bmatrix} \hat{\boldsymbol{H}} & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \ddots & ... & \mathbf{0} \\ \vdots & & \hat{\boldsymbol{H}} & \mathbf{0} \end{bmatrix} \tag{4.14}$$

where each $\hat{\boldsymbol{H}}$ matrix has dimensions $N_S \times \bar{N}$ as in Eq. (4.10), while the last column of 0-matrices ($N_S \times N_\delta$ dimensions) are due to the fact that the data-driven part is independent on the internal parameters vector $\boldsymbol{\delta}$. Moreover, the product between the $\hat{\boldsymbol{L}_{\boldsymbol{DD}}}$ and the vector of internal parameters $\boldsymbol{\theta}$ is a vector $(N_S \times N) \times 1$ containing the network outputs ordered in this way:

$$[O_{11}, ..., O_{1i}, ..., O_{1N_S}, ..., O_{NN_S}]' \tag{4.15}$$

where the first subscript stands for the output neuron number and the second one for the input sample number the output came from.
The physics-driven matrix $\hat{\boldsymbol{L}}_{\boldsymbol{PD}}$, once it is multiplied by the vector of internal parameters, returns row-by-row the expressions of the equilibrium equations residuals $\boldsymbol{R}$ and the boundary conditions residuals $\boldsymbol{B}$ evaluated at the collocation points.
The latter can be written in a generic way as follows:

$$\begin{cases} \boldsymbol{R}(\boldsymbol{\delta}, \boldsymbol{X}_{\boldsymbol{CD}}, \boldsymbol{O}) = 0 & \text{in } \Omega \\ \boldsymbol{B}(\boldsymbol{\delta}, \boldsymbol{X}_{\boldsymbol{CB}}, \boldsymbol{O}) = 0 & \text{in } \partial\Omega \end{cases} \tag{4.16}$$

where Ω is the system domain and $\partial\Omega$ is its boundary, $\boldsymbol{X}_{\boldsymbol{CD}}$ is the set of collocation points in the domain and $\boldsymbol{X}_{\boldsymbol{CB}}$ is the set of collocation points along the boundaries of the domain.
As reported in Eq. (4.16) the residuals depends on network outputs $\boldsymbol{O}$ (and, as a consequence, on output weights) and on the unknown parameters $\boldsymbol{\delta}$. The dependency on the internal parameters, generally, is non-linear, which means that $\hat{\boldsymbol{L}_{\boldsymbol{PD}}} = \hat{\boldsymbol{L}_{\boldsymbol{PD}}}(\boldsymbol{\theta})$.
Moreover, differently from the data-driven part, the $\hat{\boldsymbol{L}_{\boldsymbol{PD}}}$ is a full matrix with dimensions $N_c \times (N \times \bar{N} + N_\delta)$. It follows that the $\hat{\boldsymbol{L}}$ matrix can be written as:

$$\hat{\boldsymbol{L}} = \begin{bmatrix} \hat{\boldsymbol{L}}_{\boldsymbol{DD}} \\ \hat{\boldsymbol{L}}_{\boldsymbol{PD}}(\boldsymbol{\theta}) \end{bmatrix} \tag{4.17}$$

such that it depends on the internal parameters too.

Also the known outputs vector $\boldsymbol{F}$ can be split in two parts:

$$\boldsymbol{F} = [\boldsymbol{F}_{DD}, \boldsymbol{F}_{PD}]^T. \tag{4.18}$$

where $\boldsymbol{F_{DD}} = (U_{L11}, ..., U_{L1N_S}, ..., U_{Lij}, ..., U_{LN1}, ..., U_{LNN_S})^T$ for $i = 1, ..., N$ and $j = 1, ..., N_S$, while $\boldsymbol{F_{PD}} = (0, ..., 0)^T$ ($N_C \times 1$ dimensions, where N_C is the number of collocation points).

Since $\hat{\boldsymbol{L}}$=$\hat{\boldsymbol{L}}(\theta)$, the solution of Eq. (4.13) must be found through an iterative process. Firstly, the Jacobian $\hat{\boldsymbol{J}}_r$ of the residual $\boldsymbol{r}$=$\hat{\boldsymbol{L}}(\boldsymbol{\theta})\boldsymbol{\theta} - \boldsymbol{F}$ is evaluated for $\boldsymbol{\theta} = \boldsymbol{\theta_i}$. Then, the vector of internal parameters is updated as $\boldsymbol{\theta_{i+1}} = \boldsymbol{\theta_i} + \boldsymbol{\Delta\theta_i}$, where $\boldsymbol{\Delta\theta_i}$ is the solution of the following least square problem:

$$\hat{\boldsymbol{J}}(\boldsymbol{\theta_i})\boldsymbol{\Delta\theta_{i+1}} = \boldsymbol{r}(\boldsymbol{\theta_i}), \quad \text{for } i = 0, ..., convergence \tag{4.19}$$

In particular $\boldsymbol{\theta_0}$ is the initial guess for the internal parameters vector in the iterative process and it is user-defined.
The iterative process ends when the difference of the residual vector norm $\boldsymbol{r}$ between two consecutive iterations goes under a fixed tolerance or when the number of iterations go beyond a maximum value.

5 | Kharunen-Loeve expansions and Physics-Informed neural networks for the detection of defects

Karhunen-Loeve expansions and Physics-Informed neural networks are the fundamental tools for the development of the proposed method for defects detection.
In this chapter, it is illustrated how these tools contribute to the implementation of the method, highlighting how it has to be used and its advantages and disadvantages with respect to the previously cited techniques. The method flow-chart is depicted in Figure 5.1 and it is explained in the next section.

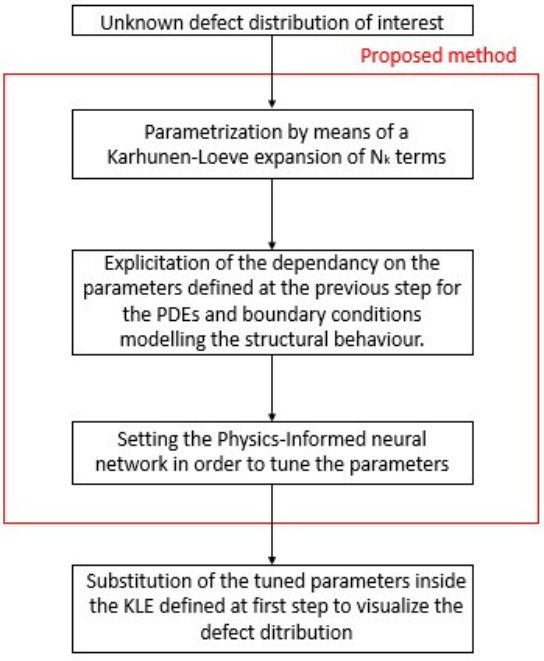


Figure 5.1: Proposed method flow-chart

5.1. Outline of the method

Defects distributions in structures are deterministic quantities, but, since their real aspects are unknowns before the detection, they can be assumed as stochastic fields.
As illustrated in Section 3.2.2., a stochastic quantity can be expressed by means of a Karhunen-Loeve expansion. Therefore, assumed $d(\boldsymbol{x})$ as the defect distribution of interest (i.e. uncertain material Young modulus in each point of the structure), whose correlation function, correlation length, mean value, variance are known in good approximation, it can be expressed similarly to Eq. (3.5):

$$d(\boldsymbol{x}, \alpha) = d_m(\boldsymbol{x}) + \sum_{i=1}^{N_k} \xi_i(\alpha) \sqrt{\lambda_i} \phi_i(\boldsymbol{x}). \tag{5.1}$$

where $d_m(\boldsymbol{x})$ is the mean value, $\boldsymbol{x}$ stands for the vector of spatial coordinates and N_k is the number of terms in the expansion.
It is straightforward that the more terms in the expansion, the better is the approximation made by the KLE of the stochastic field. There is no limit to the value of N_k, made exception for the computational effort.
After this step, the defect distribution of interest is parameterized by means of unknown variables, gathered in the vector $\boldsymbol{\Xi} = (\xi_1, ..., \xi_i, ..., \xi_{N_k})$, which are the only unknowns terms in Eq. (5.1).

The discovering of the parameters ξ_i describing the defect distribution can be thought as the solution of an inverse problem. As a matter of fact, since the presence of defects causes alterations in the mechanical performance of the structure, it is possible to make explicit the dependency of its elastic properties on those parameters. This means that the sets of equilibrium equations ($\boldsymbol{R_D}$) and boundary conditions ($\boldsymbol{B_D}$) modelling the structural behaviour, which depend on the elastic properties, can be expressed as a function of $\boldsymbol{\Xi}$:

$$\begin{cases} \boldsymbol{R_D}(\boldsymbol{\Xi}, \boldsymbol{x}) = \boldsymbol{0}, & \boldsymbol{x} \in \Omega \\ \boldsymbol{B_D}(\boldsymbol{\Xi}, \boldsymbol{x}) = \boldsymbol{0}, & \boldsymbol{x} \in \partial\Omega \end{cases} \tag{5.2}$$

A significant example regarding this step can be done considering the gaps and overlaps affecting a composite plate manufactured by means of AFP process. As a matter of fact, the presence of these defects can be assumed as an uncertain distribution of the fibers volumetric fraction V_f in the structure. Once the volumetric fraction is parameterized as in Eq. (5.1), its expression can be substituted into Eq. (2.1) (considering that $V_m =$

$1-V_f$). It follows that the elastic properties of the plate, point by point, are parameterized according to the same variables used for the parameterization of V_f.
In particular, if the plate is loaded in bending by a distributed force p (see Figure 5.2)

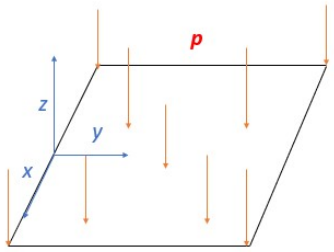


Figure 5.2: Plate loaded in bending

and the Kirchhoff assumptions are adopted, the equilibrium of forces along the Z-direction implies:

$$\boldsymbol{\Delta M} = p \tag{5.3}$$

where $\boldsymbol{\Delta} = (\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, 2\frac{\partial^2}{\partial x \partial y})$ is the vector of partial derivatives and $\boldsymbol{M}$ is the vector of moments per unit length introduced in Section 2.1.2.. From Eq. (2.4), $\boldsymbol{M}$ can be written as $\boldsymbol{M} = [\hat{\boldsymbol{B}}\hat{\boldsymbol{D}}]\boldsymbol{\epsilon}$, where the matrices $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{D}}$ depends on material elastic properties. It follows that also Eq. (5.3) can be written as function of the variables parametrizing V_f. After this step, the structural underlying model expression has been organized in order to highlight the dependency on the parameters defining the vector $\boldsymbol{\Xi}$. In order to discover the latter, it is necessary to solve an inverse problem.

As illustrated in Section 4.2, a tool widely used to solve structural inverse problem is the Physics-informed neural network. In the proposed method, it is used together with the ELM for the training process.
The input of the network is the vector $\boldsymbol{x}$, which collects the coordinates of the points in the domain. The output is the displacements vector $\boldsymbol{u}$. The vector $\boldsymbol{\Xi}$ is included in the internal parameters vector $\boldsymbol{\theta}$, such that it can be tuned during the training process.
Since the number of neurons for the three layers (input, hidden and output layers), the internal parameters, the collocation points and the labeled points are problem depending, in Figure 5.2 is depicted only a generic architecture of the neural network used in the method:

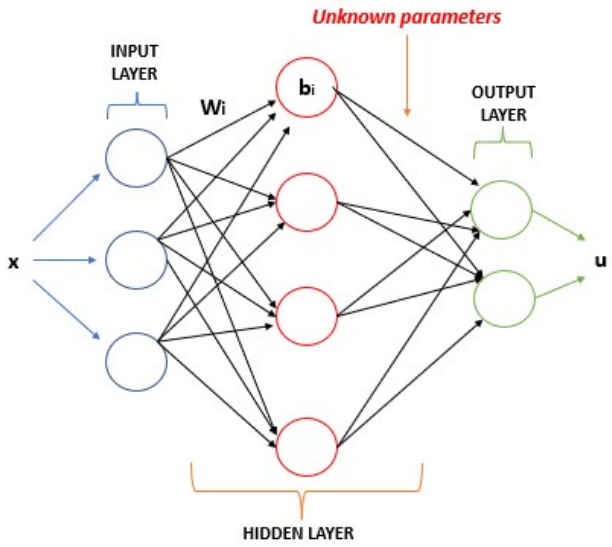


Figure 5.3: Generic network architecture for the proposed method

At the end of the training process, the vector Ξ is known and, so, its components can be substituted inside the Eq. (5.1). Doing so, an analytical expression of the defect distribution is obtained.

The above description is made considering one unknown defect distribution in the structure, but the proposed method can be applied also in case of multiple unknown distributions.
The steps are the same depicted in Figure 5.1, remembering that each defect distribution must be parameterized using a different set of parameters.

5.2. Remarks

In the introduction to the thesis, some techniques exploiting the neural networks for defects detection were briefly illustrated [8, 27, 28, 35]. However, some critical aspects could be found in them.
The necessity to exploit other techniques (like in [8, 27]), which could be costly or could require skilled personnel for the set-up and the interpretation of the results, to obtain the labeled data for the training process is one of them.
It is not always provided the location of the defect. As example, in [35] the output of the network is a number representing the quality of the weld bead, but no information is provided regarding the location of defects arised during the welding process.
Moreover, the network output could be not easy to be interpreted [28].

On the contrary, the proposed method returns analytical expressions of the defect distributions at the end of the training process, such that locations and entities of the defects are completely known. Moreover, the training process requires only the displacements at a set of labeled points of the structure (under a certain loads and boundary conditions), which is an easily accessible information from a laboratory activity.
Secondly, even if the proposed method was thought for VAT structures, it can potentially work for any type of structure, metallic or composite, and any kind of defects.

The critical aspect is that it is necessary to choose, in a good manner, the correlation function, correlation length and variance of the stochastic field used to represent the defect distribution of interest in Eq. (5.1). As a matter of fact, these could be information not easily available.

As depicted in Figure 5.4, the proposed method is thought to be the intermediate step between the real structure and its numerical model.
As a matter of fact, the analytical expressions that are obtained from the method can be employed to build any kind of numerical model (Ritz, FEM and others). This allows to do a mechanical characterization of the structure without the necessity to make experimental activity on it, saving time and economical resources.

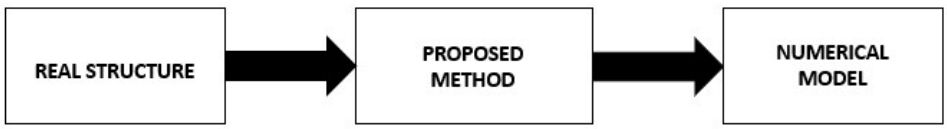


Figure 5.4: Proposed method generic application flowchart

6 | Applications and results

To prove the reliability of the proposed method, different types of problems were tackled, starting from the detection of the material Young modulus distribution for a rod, to the determination of fiber misalignment and volumetric fraction distributions for a VAT plate.
Once the problem to address was well-defined, the generic steps followed to test the effectiveness of the method were:

1. the perturbation of the nominal distribution of one or more structural properties (i.e. material Young modulus, fibers orientation, volumetric fraction) according to known Kharunen-Loeve expansions. The result was a defected structure characterised by known defects distributions;
2. the calculation of the displacement field of the defected structure to be used for the training process of the network;
3. the set-up of neural network in terms of number of layers neurons, labeled points, collocation points, arrest criteria characteristics (maximum number of iterations and tolerance) and internal parameters to be tuned. Moreover, equilibrium equations and boundary conditions residuals were defined;
4. the comparison between the known distributions at step 1 and the corresponding obtained at the end to the training process.

The results were obtained using a Physics-Informed Neural Network, implemented in a MATLAB environment, trained employing the Extreme Learning Machine algorithm.
Moreover, the structural properties were perturbed considering them as Gaussian processes. The perturbation was done by means of Kharunen-Loeve expansions built considering exponential correlation functions and plausible values for correlation length and variance. In particular, the latter were built by means of Eq. (3.8), Eq. (3.9) in case of one dimensional problems, while by means of Eq. (3.12) and Eq. (3.13) in case of the two dimensional ones.

In this chapter, the results obtained applying the proposed method are reported. In particular, for each problem, it is provided its description, how the labelled data are obtained, the set-up of the neural network and a critical analysis of the corresponding results.
Different kinds of problems with increasing level of complexity are considered.

6.1. Detection of material Young modulus for a rod

The first addressed problem regards the detection of the material Young modulus distribution for a rod.
As depicted in Figure 6.1, the rod has length L, constant cross section area A, it is loaded by a concentrated force F at one end and it is clamped to the other.

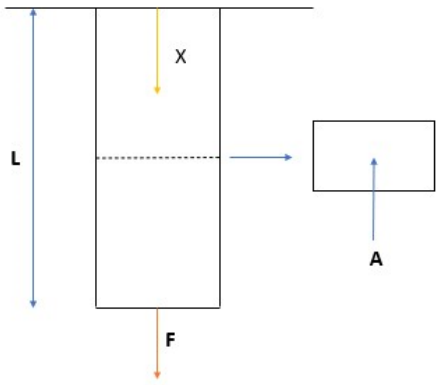


Figure 6.1: Representation of the first problem

From the equilibrium of an infinitesimal segment of the rod, the structural model is described by the following PDE and boundary conditions:

$$\begin{cases} \dfrac{d}{dx}(E(x)A(x)\dfrac{d}{dx}u(x)) = 0, & \text{for } 0 \leq x \leq L \quad (6.1a) \\ u(0) = 0 \quad (6.1b) \\ E(L)A(L)\dfrac{d}{dx}u(L) = F \quad (6.1c) \end{cases}$$

where $E(x)$ is the Young modulus distribution and $u(x)$ is the axial displacement.

Starting from a constant Young modulus distribution $E(x) = const$, the latter is perturbed

by means of a generic one-dimensional Karhunen-Loeve expansion, such that:

$$E_P(x) = E(x) + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_{Pi}. \tag{6.2}$$

Following [39] the number of terms in the expansion is limited to ten.
As example, in Figure 6.2 it is depicted a possible perturbed Young modulus distribution $E_P(x)$ considering $E(x) = 70000MPa$, a variance $\sigma^2 = 20000MPa^2$ and correlation length $c = L$.

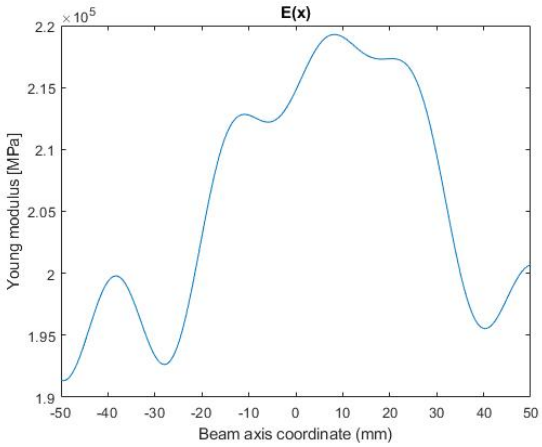


Figure 6.2: Example of perturbed Young modulus distribution $E_P(x)$

This process is done by means of a MATLAB subroutine, which takes as inputs the variance σ^2 and the correlation length c, while it gives as output a possible perturbed distribution.
Since, by definition (see Section 3.2.2.), the ten variables ξ_{pi} in Eq. (6.2) must be random and uncorrelated, they are sampled by means of Hypercube Latin sampling.

6.1.1. Generation of labeled inputs and outputs sets

The training of the neural network requires the labeled inputs set $\boldsymbol{X_L} = (X_{L1}, ..., X_{LN_S})$ and labeled outputs set $\boldsymbol{O_L} = (O_{L1}, ..., O_{LN_S})$.
As written in Section 5.1., the inputs set contains the X-axis coordinate of N_S points (labeled points), while the outputs set the correspondents X-axis displacements.
In particular, considering the geometric properties, loads, and boundary conditions men-

tioned before, the axial displacement of the rod can be found as it follows:

$$u(\alpha) = \frac{F}{A}\int_0^{\alpha} \frac{1}{E_P(x)}dx \quad 0 \leq \alpha \leq L. \tag{6.3}$$

where α is a generic point along the beam axis.
Tha analytical solution of Eq. (6.2) requires the integration of the inverse of the perturbed Young modulus field. Due to the complexity of its analytical expression, the integration is performed numerically.

Moreover, the inputs set is mapped into an dimensionless domain $D_a : [-1; 1]$ because it is advantageous from a neural network performances point of view. This is done by means of a linear transformation, such that the new inputs set coordinates are $Xa_i = -1 + \frac{X_i}{L}$ for i=1,...,N_S.

6.1.2. Neural network set-up

The first step is the definition of residuals for the PDE and for the boundary conditions describing the behaviour of the system under investigation.
From Eq. (6.1a) it follows that the PDE residual, which is evaluated in a generic collocation point X_{CDi} (for $i = 1, ..., N_{CD}$), is:

$$R(X_{CDi}, \boldsymbol{\theta}) = \frac{d}{dx}[E_N(X_{CDi})A\frac{d}{dx}u_N(X_{CDi})] = 0, \quad \text{for } i = 1, ..., N_{CD}. \tag{6.4}$$

where $u_N(x)$ is the output of the neural network evaluated at the collocation point X_{Ci}, $E_N(x)$ is the network approximation of the Young modulus distribution and $\boldsymbol{\theta}$ is the vector of internal parameters.
The residual of the geometric boundary condition is:

$$B_G(\boldsymbol{\theta}) = u_N(0) = 0. \tag{6.5}$$

The residual regarding the natural boundary condition is:

$$B_N(\boldsymbol{\theta}) = E_N(1)A\frac{d}{dx}u_N(1) - F = 0. \tag{6.6}$$

$E_N(x)$ is the Young modulus distribution approximation made by the neural network during the training process. In particular, it is parameterized by means of a KLE with the same stochastic properties (covariance function, variance and correlation length) and number of terms used for the perturbation of $E(x)$. It follows that $E_N(x)$ is:

$$E_N(x) = E(x) + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i\xi_{Ni}. \tag{6.7}$$

The ten ξ_{Ni} variables are the unknown parameters in Eq. (6.7) and they are included in the internal parameters vector $\boldsymbol{\theta}$ in order to be tuned during the training process.
It is worth mentioning that, since the network is trained through dimensionless inputs, also the KLE in Eq. (6.7) must be mapped in the dimensionless domain D_a.

The architecture of the network is depicted in Figure 6.3.

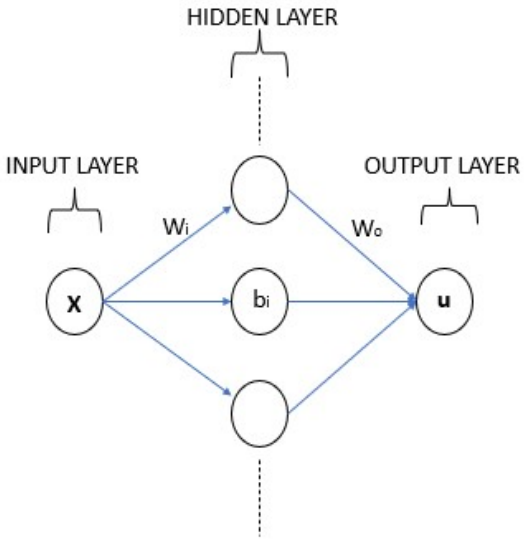


Figure 6.3: Generic neural network architecture for 1D problems

It is a network that takes in input the X-axis coordinate (one neuron for the input layer) and returns in output the approximation of the corresponding axial displacement (one neuron for the output layer). The number of neurons in the hidden layer, instead, is equal to 2000.

The output weights and the variables in Eq. (6.7) are the internal parameters learned during the training process, for a total number of 4010. In particular, during the training process the network uses N_S=100 labeled points and N_{CD}=200 collocation points.
Moreover, the maximum number of iterations is equal to 15 and the value of tolerance for the norm of the residual is fixed to 10^{-5} for the training process to be arrested.
The initial guess for the output weights is a random value between -1 and 1, while for the other internal parameters the initial value is set to 0 such that the initial guess for $E_N(x)$ is $E(x)$.

6.1.3. Results

The following results refers to a rod of length L=100 mm, section area A= 100 mm^2 and loaded by a concentrated force F=1000 N. The Young modulus has been perturbed using the following stochastic properties:

- Average value $E(x)$=70000 MPa;
- correlation length c equal to the beam length;
- Correlation function $C(x_1, x_2)$=$e^{-\frac{|x_1-x_2|}{c}}$;
- Variance σ^2=2000 MPa^2

Using the subroutine described in Section 6.1, the deterministic distribution $E_P(x)$ assigned to the rod is represented in Figure 6.4.

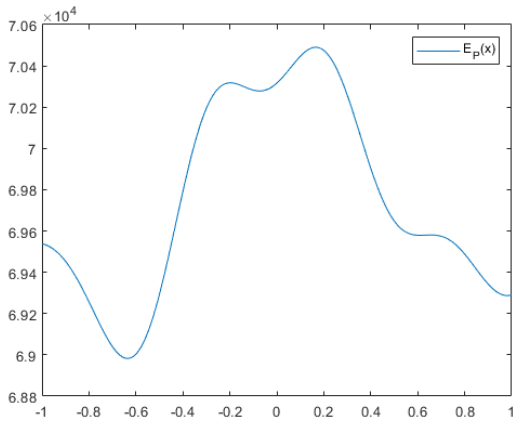
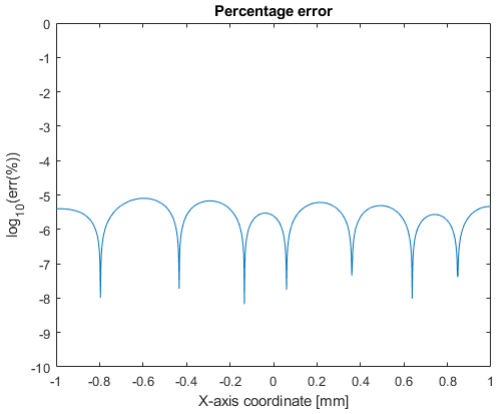


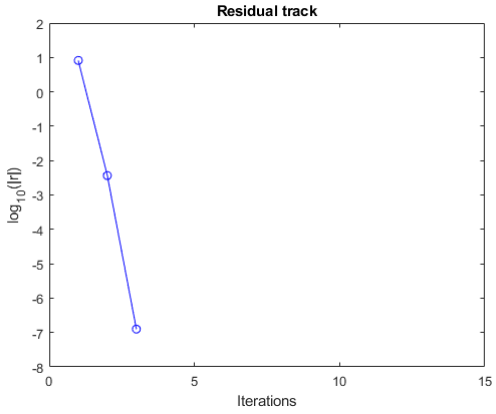
Figure 6.4: Young modulus distribution assigned to the rod

The neural network is trained for each activation functions cited in Section 4.1. In Figure

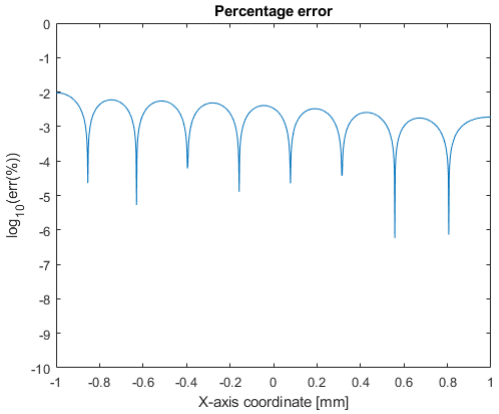
6.5 it is reported the distribution of the final percentage error $err\%(x) = |\frac{E_N(x)-E_P(x)}{E_P(x)}| \cdot 100$ and the evolution of the norm of the residual vector $\boldsymbol{r}$, both in logarithmic scale.



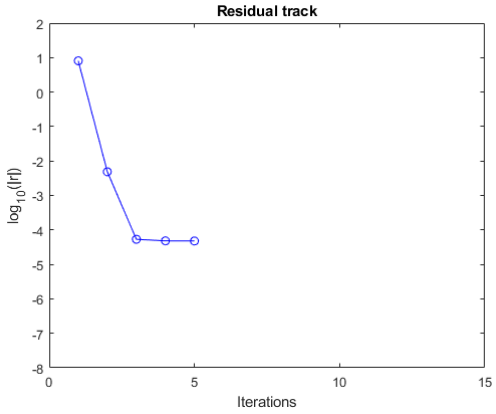
(a) Percentage error using *atan*



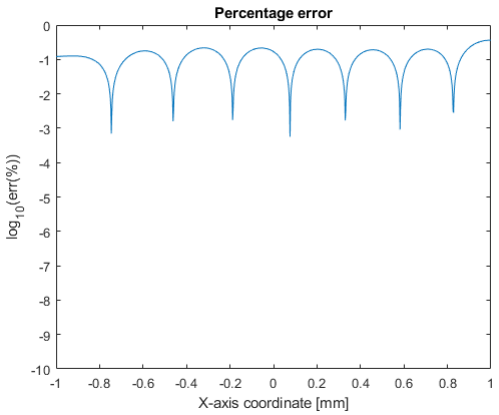
(b) Residual norm track using *atan*



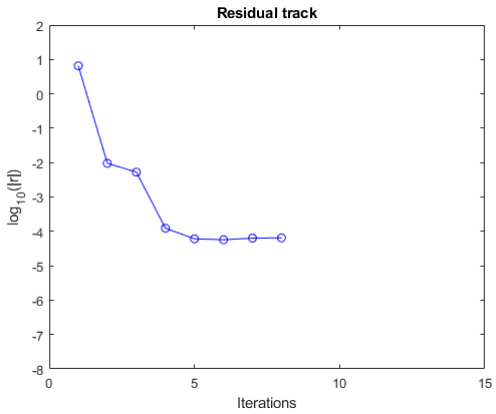
(c) Percentage error using *tanh*



(d) Residual norm track using *tanh*



(e) Percentage error using *sigmoid*

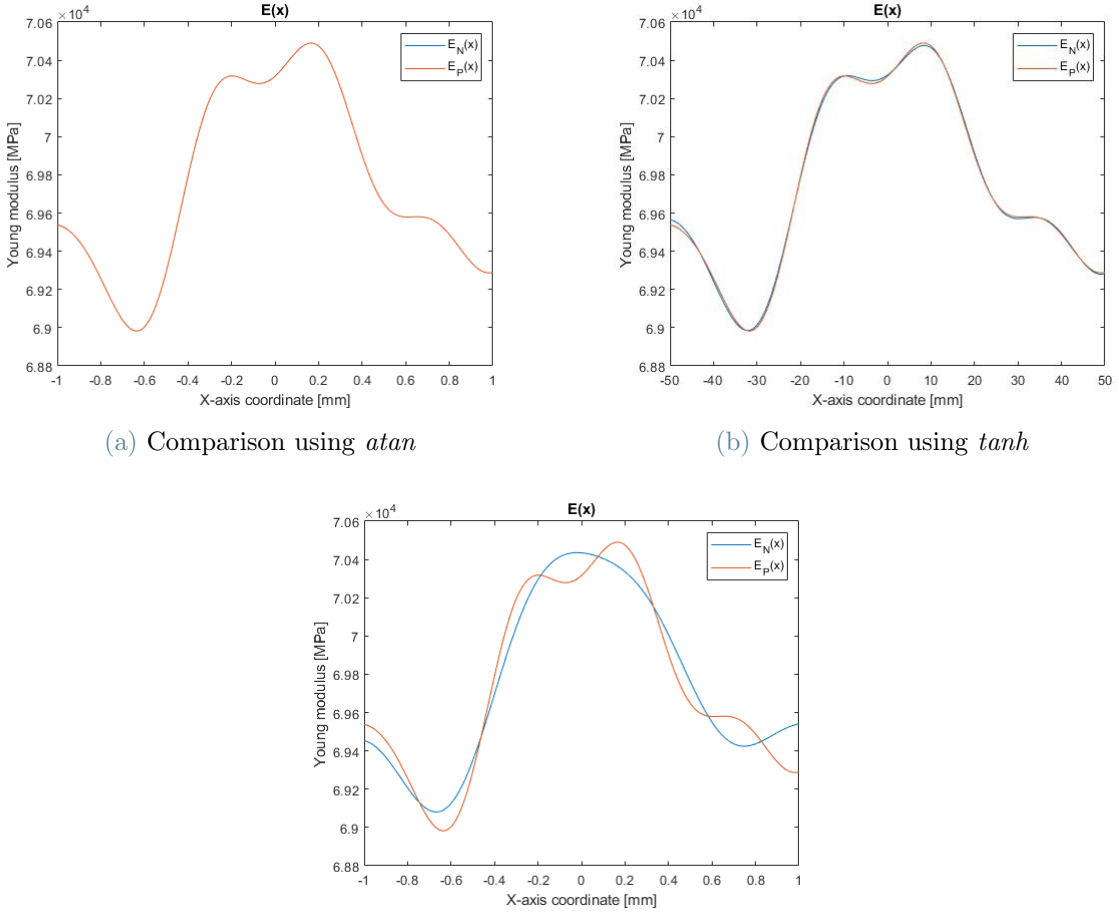


(f) Residual norm track using *sigmoid*

Figure 6.5: Final percentage error and residual vector norm evolution

As seen, the network is able to reconstruct very well the Young modulus distribution using *atan* and *tanh*, since the percentage error are in the order of 10^{-6} and 10^{-3}, respectively. Regarding the *sigmoid*, indeed, the percentage error is a little bit higher, in the order of 10^{-1}.
Moreover, when using *sigmoid* the number of iterations to reach convergence is higher with respect to the others.

By doing a comparison of the results obtained in terms of $E_N(x)$vs.$E_P(x)$ (see Figure 6.6), it is possible to notice a feature.



(a) Comparison using *atan*

(b) Comparison using *tanh*

(c) Comparison using *sigmoid*

Figure 6.6: $E_N(x)$(in blue)vs.$E_P(x)$(in red)

As a matter of fact, it seems that the network has difficulty in achieving good results when using *sigmoid*, especially in correspondence of rapid variation of the second derivatives of

the Young modulus distribution. This trend is confirmed by the results of several analyses involving different perturbed distributions $E_P(x)$. Neither changing the number of hidden neurons, nor increasing the number of collocation points could solve this issue.

6.1.4. Bending of beam

Herein it is reported the application of the method for a beam loaded in bending. The purpose is to show how the quality of the results changes when different equilibrium equations and boundary conditions are considered.
As depicted in Figure 6.7, the beam is clamped and loaded by a concentrated shear force F at the tip.

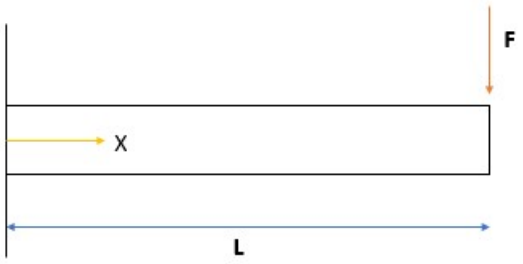


Figure 6.7: Bending load condition

Adopting the Euler-Bernoulli hypothesis, from the Principle of Virtual Work (PVW) the equilibrium equation is:

$$\frac{d}{dx}[\frac{d}{dx}E(x)J\frac{d^2}{dx^2}w(x) + E(x)J\frac{d^3}{dx^3}w(x)] = 0 \tag{6.8}$$

while the natural boundary conditions are:

$$\begin{cases} \frac{d}{dx}[E(L)J\frac{d^2}{dx^2}w(L)] + F = 0 & \text{(6.9a)} \\ \frac{d}{dx}[E(L)J\frac{d}{dx}w(L)] + F = 0 & \text{(6.9b)} \end{cases}$$

where J is the cross section intertia and $w(x)$ is the vertical displacement.
Then, the structural model is completed by the following geometric boundary conditions:

$$\begin{cases} w(0) = 0 & \text{(6.10a)} \\ \frac{d}{dx}w(0) = 0 & \text{(6.10b)} \end{cases}$$

The neural network set-up is done as before.
The PDE residual to be evaluated at the collocation points is:

$$R_1(X_{Ci}, \boldsymbol{\theta}) = \frac{d}{dx}[\frac{d}{dx}E_N(X_{Ci})J\frac{d^2}{dx^2}w_N(X_{Ci}) + E_N(X_{Ci})J\frac{d^3}{dx^3}w_N(X_{Ci})] = 0 \tag{6.11}$$

The boundary conditions residuals are:

$$\begin{cases} B_{G1}(\boldsymbol{\theta}) = w(0) = 0 \\ B_{G2}(\boldsymbol{\theta}) = \dfrac{d}{dx}w(0) = 0 \\ B_{N1}(\boldsymbol{\theta}) = \dfrac{d}{dx}[E_N(1)J\dfrac{d^2}{dx^2}w(1)] + F = 0 \\ B_{N2}(\boldsymbol{\theta}) = \dfrac{d}{dx}[E_N(1)J\dfrac{d}{dx}w(1)] + F = 0 \end{cases} \tag{6.12}$$

Moreover, the vertical displacement to be included in the labeled outputs set $\boldsymbol{O_L}$ can be found by solving this equation:

$$w(\alpha) = \frac{F}{J}\int_0^{\alpha} \frac{(L-x)(\alpha-x)}{E_P(x)}dx \quad 0 \leq \alpha \leq L. \tag{6.13}$$

where α is a generic point along the beam axis.
The final architecture of the network is the same reported in Figure 6.3 and described in Section 6.1.2.

The results below are referred to a beam with the same geometric properties and Young modulus stochastic properties used in the previous case, while the applied load has a magnitude of 100 N.
In order to make a reliable comparison with the results obtained in Section 6.1.3., the perturbed Young modulus distribution $E_P(x)$ is the same as before, as well as the initial guesses for the training process and the arrest criteria features.
The most representative results from the training process are summarized in Table 6.3. In particular, the order of the final percentage error distribution $err\%(x) = |\frac{E_N(x)-E_P(x)}{E_P(x)}|\cdot 100$, the number of iterations and the running time (normalized with respect to one from the *atan* case) are reported.

Act.function	Percentage Error Order	Iterations	Running time
ATAN	10^{-3}	5	1
TANH	10^{-1}	10	1.88
SIGMOID	10^{0}	10	1.87

Table 6.1: Main results

It is possible to notice that, for each activation function, the percentage error order is higher with respect to the previous case. This can be addressed to the maximum order of derivative of the vertical displacement in Eq. (6.8), which is two times higher with respect to the one in Eq. (6.1a) for $u(x)$. As a matter of fact, since the network is trained using only the displacements, there is no guarantee that the derivatives are approximated properly (and the situation gets worse increasing the order of the derivative), as highlighted also in [49].
Moreover, the number of iterations necessary for the convergence is higher, as well as computational time.

6.2. Detection of section area and Young modulus for a rod

In Section 6.1 the problem is characterised by only one uncertain structural property (the material Young modulus). In real applications, it makes sense to think that more uncertainties coexist in the structure. For this reason, the purpose in this section is to assess the ability of the method to detect and distinguish the uncertain distributions of more than one structural property at the same time.
In particular, starting from a constant Young modulus and section area, respectively $E(x)$ and $A(x)$, they are perturbed by means of different KLEs, made of ten terms each one. Then, these perturbed distributions must be learned by the neural network through the training process.

The labeled outputs set $\boldsymbol{O_L}$ for the training process contains the axial displacement at N_S points and it is built by solving the following integral:

$$u(\alpha) = F \int_0^{\alpha} \frac{1}{E_P(x) \cdot A_P(x)} dx \quad 0 \leq \alpha \leq L \tag{6.14}$$

where $E_P(x)$ and $A_P(x)$ are the perturbed distributions:

$$\begin{aligned} E_P(x) &= E_i + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_{Pei} \\ A_P(x) &= A_i + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_{Pai} \end{aligned} \tag{6.15}$$

Moreover, the inputs set containing the 1D spatial coordinates of the labeled points is mapped into the dimensionless domain D_a, exploiting the same linear transformation used in Section 6.1.

6.2.1. Neural network set-up

The PDE residual, which is evaluated at a set of N_{CD} collocation points X_{Ci}, is:

$$R(X_{Ci}, \boldsymbol{\theta}) = \frac{d}{dx}[E_N(X_{Ci})A_N(X_{Ci})\frac{d}{dx}u_N(x)], \quad \text{for } i = 1, ..., N_{CD} \tag{6.16}$$

The boundary conditions residuals are:

$$\begin{cases} B_G(\boldsymbol{\theta}) = u_N(0) = 0 \\ B_N(\boldsymbol{\theta}) = \dfrac{d}{dx}[E_N(1)A_N(1)\dfrac{d}{dx}u_N(1)] + F = 0 \end{cases} \tag{6.17}$$

In particular, $E_N(x)$ and $A_N(x)$ are the neural network approximations of $E_P(x)$ and $A_P(x)$, respectively. They are expressed by means of the KLEs, using the same stochastic properties used for the perturbation. It follows that:

$$\begin{aligned} E_N(x) &= E_i + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_{ei} \\ A_N(x) &= A_i + \sum_{i=1}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_{ai} \end{aligned} \tag{6.18}$$

where the variables ξ_{ei} and ξ_{ai} are the unknown parameters to be tuned during the training process.

The comparison between the vertical displacements at the labeled points and the evaluation of the residuals at the collocation points are not enough to obtain acceptable results from the training process. As a matter of fact, it has been observed that, when using only this information, the neural network is able to converge to a physical valid solution, but not to the expected one.
For a rod-like structure, the elastic properties are linear dependent with respect to its section area and Young modulus. This means that a variation of these two quantities has the same effect on structural outputs. It follows that both E and A can equally contribute to a change in the response of the structure. In addition, both variables appear in the equilibrium equations with the same order of derivative in each term.
For these reasons, the network is not able to distinguish properly these quantities. The main consequence is that the training process allows to find a solution that minimize the cost function, but different with respect to the expected one.
These considerations highlight that the network need other information to find the expected solution. In this case, the strategy proposed is to include the evaluation of the following residual at a set of N_{LS} labelled points:

$$\frac{F}{A_N(X_{LSi})} - \sigma_i = 0, \quad \text{for i=1,...,}N_{LS}. \tag{6.19}$$

where σ_i is the normal stress for the cross section in correspondence of X_{LSi}.

The architecture of the network is the same depicted in Figure 6.3. So one neuron for the input and output layer, while 1000 neurons are necessary for the hidden layer.
The internal parameters to be learned during the training process are the output weights and the parameters defining $A_N(x)$ and $E_N(x)$, for a total number of 1020.
For this purpose, the training process uses N_S=100 labeled points, N_{CD}=102 collocation points for the evaluation of PDE and boundary conditions residuals and N_{LS}=20 collocation points for the evaluation of the residual regarding the normal stress.
Moreover, the initial guess for the output weights are random values between -1 and 1, while for the other internal parameters the initial values are set to 0 such that the initial guesses for $E_N(x)$ and $A_N(x)$ are equal to $E(x)$ and $A(x)$, respectively.
The maximum number of iteration is fixed to 15, while the tolerance to 10^{-5}.

6.2.2. Results

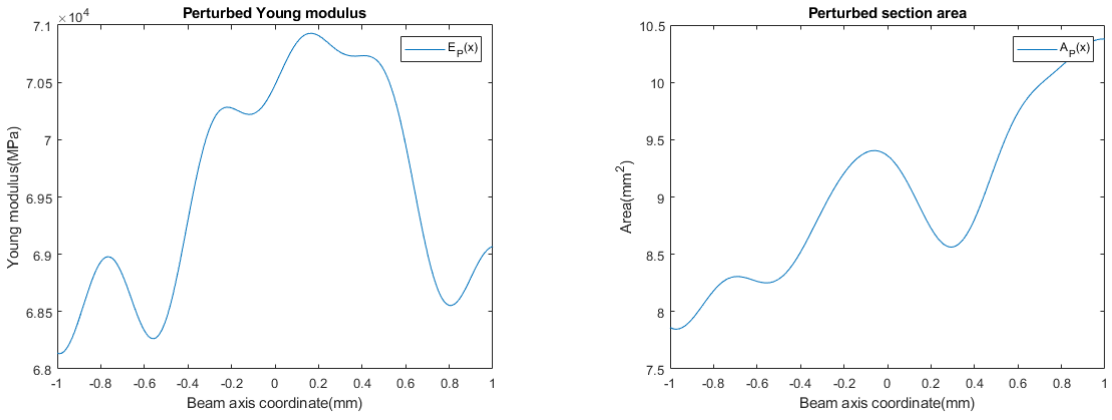
The following results refers to a beam of length L=100 mm and loaded by a concentrated force F=100 N. The Young modulus has the following stochastic properties:

- Average value E_i=70000 MPa;
- correlation length c_E equal to the beam length;
- Correlation function $C_E(x_1, x_2)$=$e^{-\frac{|x_1-x_2|}{c_E}}$;
- Variance σ_E^2=2000 MPa^2

For the section area, the following stochastic properties are considered:

- Average value A_i=10 mm^2;
- correlation length c_A equal to the beam length;
- Correlation function $C_A(x_1, x_2)$=$e^{-\frac{|x_1-x_2|}{c_A}}$;
- Variance σ_A^2= 2 mm^2

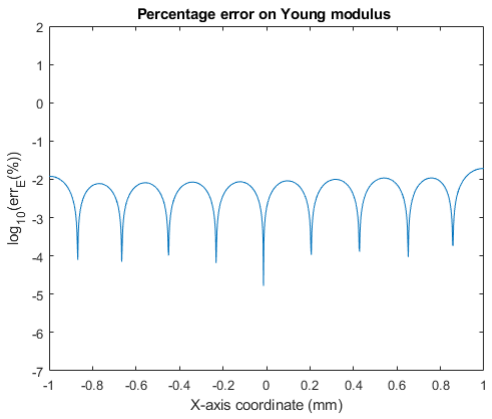
It follows that the deterministic fields, $E_P(x)$ and $A_P(x)$, assigned to the rod are depicted in Figure 6.8.



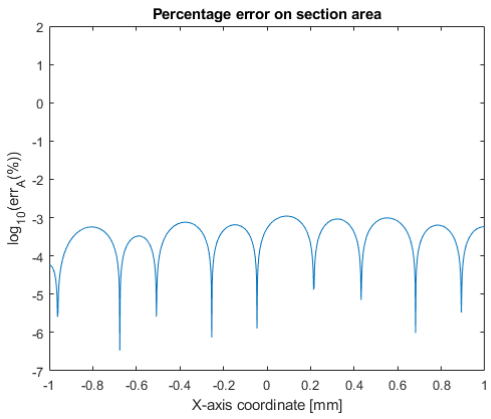
(a) $E_P(x)$ (b) $A_P(x)$

Figure 6.8: Perturbed distributions assigned to the rod

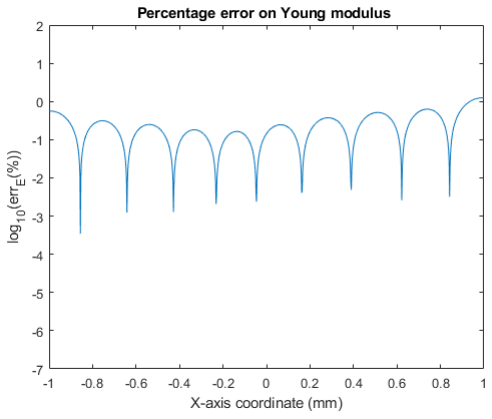
The results depicted in Figure 6.9 are the final percentage error distribution $err_E\%(x) = |\frac{E_N(x)-E_P(x)}{E_P(x)}| \cdot 100$ on the Young modulus field and the final percentage error distribution $err_A\%(x) = |\frac{A_N(x)-A_P(x)}{A_P(x)}| \cdot 100$ on the cross section area, both in logarithmic scale.



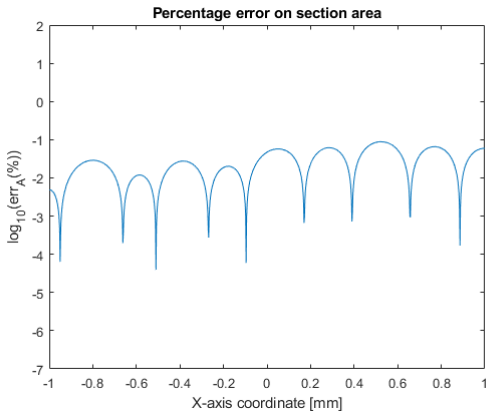
(a) Percentage error on $E(x)$ using *atan*



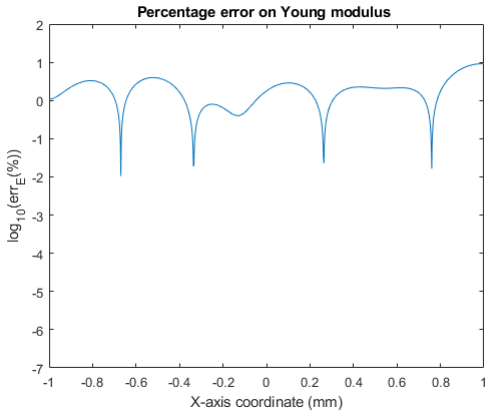
(b) Percentage error on $A(x)$ area using *atan*



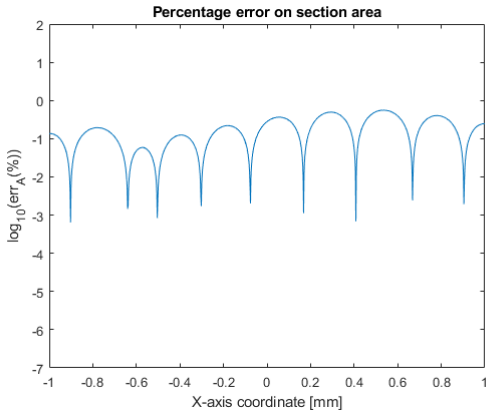
(c) Percentage error on $E(x)$ using *tanh*



(d) Percentage error on $A(x)$ using *tanh*



(e) Percentage error on $E(x)$ using *sigmoid*

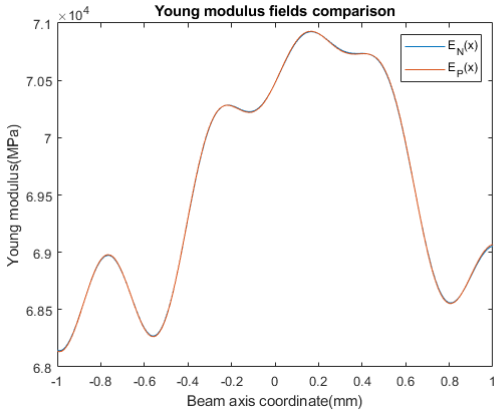


(f) Percentage error on $A(x)$ using *sigmoid*

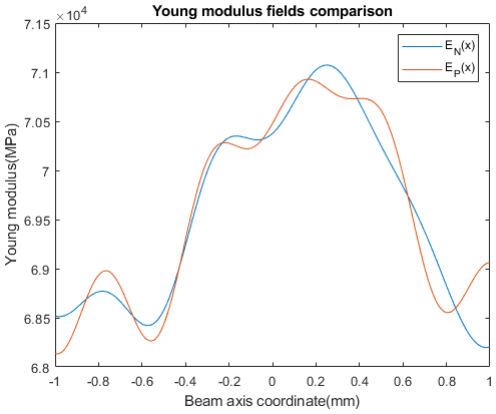
Figure 6.9: Final percentage errors

As seen, the *atan* is the activation function that provides the best results, with percentage errors in the order of 10^{-3}. These values are larger for the other two activation functions, especially for the Young modulus.

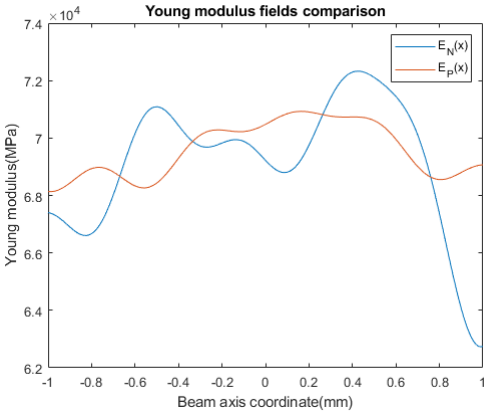
It is interesting to observe that the percentage error on the Young modulus is higher with respect to the section area one. This difference is more clear in Figure 6.10 and Figure 6.11.



(a) $E_N(x)$vs.$E_P(x)$ using *atan*

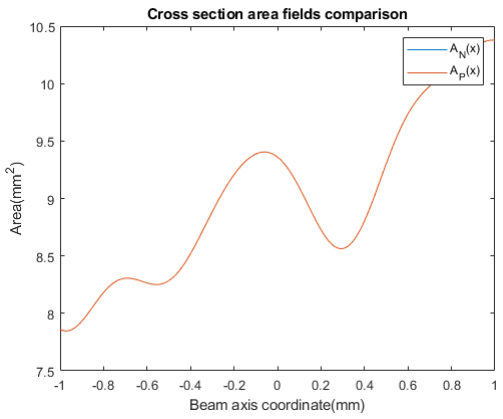


(b) $E_N(x)$vs.$E_P(x)$ using *tanh*

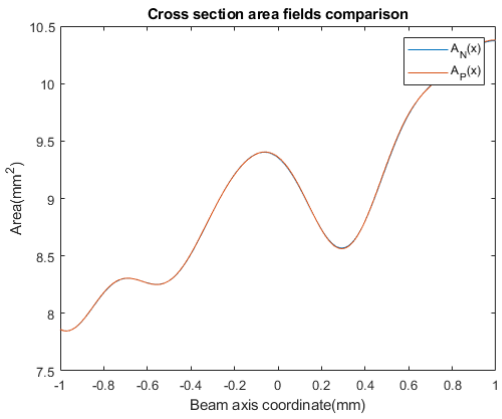


(c) $E_N(x)$vs.$E_P(x)$ using sigmoid

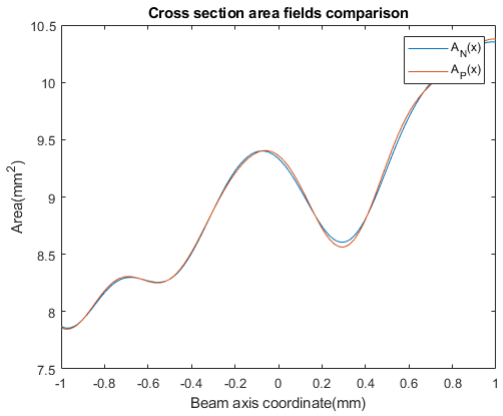
Figure 6.10: $E_N(x)$(in red)vs.$E_P(x)$(in blue)



(a) $A_N(x)$vs.$A_P(x)$ using *atan*



(b) $A_N(x)$vs.$A_P(x)$ using *tanh*

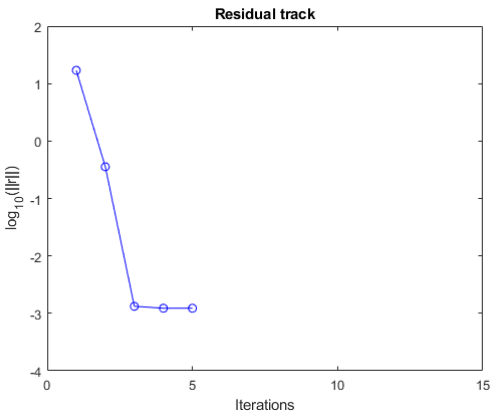


(c) $A_N(x)$vs.$A_P(x)$using *sigmoid*

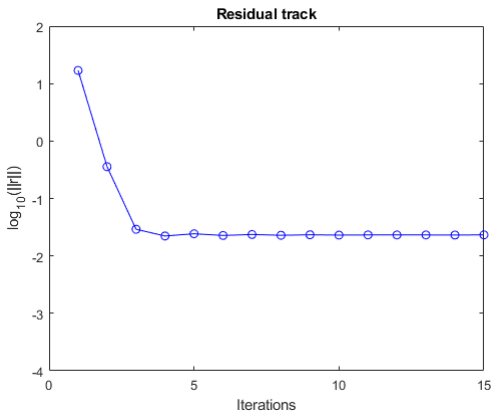
Figure 6.11: $A_N(x)$(in red)vs.$A_P(x)$(in blue)

This is due to the fact that the network is enriched with more information directly connected to the cross section area (see Eq. (6.19)).

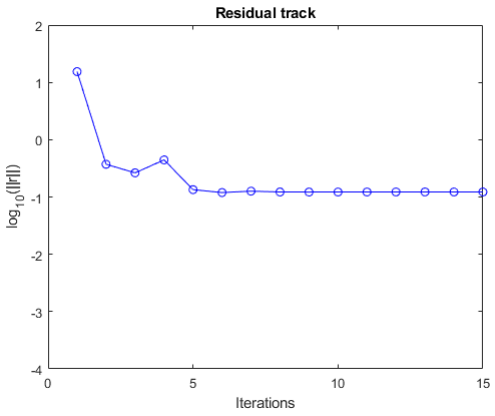
The fact that *atan* works better is also visible from Figure 6.12. As a matter of fact, the network reaches the convergence much before the other cases.



(a) Residual track using *atan*



(b) Residual track using *tanh*



(c) Residual track using *sigmoid*

Figure 6.12: Residual vector norm evolution

The convergence is not reached for *tanh* and *sigmoid* within the 15 iterations, but it is necessary to arrive to the iteration number 22 and 49 respectively.

6.3. Detection of Young modulus for an isotropic plate

The first type of two dimensional problem faced is the detection of material Young modulus distribution for a plate. In particular, in this section the purpose is to show that the proposed method can work also for detecting constant offsets of the uncertain structural properties, like the Young modulus one, with respect their nominal values. This is different from what was done in the previous sections, in which the selected initial constant distributions were perturbed by a random fields built by means of KLEs.
A square plate of thickness t, simply supported along each edge of length l and loaded by

a distributed force P, as depicted in Figure 6.13, is considered.

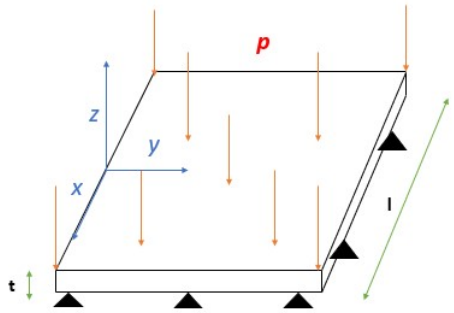


Figure 6.13: Metallic plate

Adopting the Kirchhoff hypothesis, the plate withstands the load thanks to the bending moments, such that the equilibrium equation along the Z-axis is:

$$\frac{d^2}{dx^2}M_x + \frac{d^2}{dy^2}M_y + 2\frac{d^2}{dxdy}M_{xy} = -p \tag{6.20}$$

where M_x, M_y and M_{xy} are the moments per unit length introduced in Section 2.1.2. Still from Section 2.1.2, the moments per unit length can be linked to the generalized deformations vector through the matrices $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{D}}$, but, by definition, $\hat{\boldsymbol{B}}$ is equal to zero for a plate whose elastic properties are constant through the thickness, as in this case. This means that $\boldsymbol{M} = \hat{\boldsymbol{D}}\boldsymbol{k}$, where $\boldsymbol{k}$ is the vector of curvatures $\boldsymbol{k} = (k_x, k_y, 2k_{xy})^T = (-\frac{\partial^2}{\partial x^2}w, -\frac{\partial^2}{\partial y^2}w, -2\frac{\partial^2}{\partial x \partial y}w)^T$, such that Eq. (6.20) can be written in matrix form as:

$$\begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \begin{bmatrix} \hat{D}_{11}(x,y) & \hat{D}_{12}(x,y) & 0 \\ \hat{D}_{12}(x,y) & \hat{D}_{22}(x,y) & 0 \\ 0 & 0 & \hat{D}_{66}(x,y) \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial x^2}w(x,y) \\ \frac{\partial^2}{\partial y^2}w(x,y) \\ 2\frac{\partial^2}{\partial x \partial y}w(x,y) \end{bmatrix} = p \tag{6.21}$$

where w is the out-of-plane displacement and $\hat{\boldsymbol{D}}$ matrix is:

$$\hat{\boldsymbol{D}} = \frac{E(x,y)t^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{6.22}$$

The structural model is completed with the following boundary conditions:

$$\begin{cases} w = 0 \quad \text{for } x = 0, l \text{ and } y = 0, l \\ \dfrac{d^2}{dx^2} w = 0 \qquad\qquad \text{for } y = 0, l \\ \dfrac{d^2}{dy^2} w = 0 \qquad\qquad \text{for } x = 0, l \end{cases} \tag{6.23}$$

The initial Young modulus constant distribution $E(x, y) = const$ is perturbed by means of a constant offset ΔE, such that $E_P(x, y) = E(x, y) + \Delta E$.
The neural network training process has the purpose to quantify this offset. In this regard, the approximation of the Young modulus $E_N(x, y)$ is expressed by means of a generic 2D Karhunen-Loeve expansion, whose variables have to be learned during the training process. This is necessary to simulate a real case, where the distribution of Young modulus has to be assumed to be random and not constant.

6.3.1. Generation of labeled inputs and outputs sets

An important step for the solution of the problem is the generation of the labeled inputs set $\boldsymbol{X_L}$ and the corresponding outputs set $\boldsymbol{O_L}$.
The inputs set contains the spatial coordinates of N_S labeled points, such that each sample is a pair of spatial coordinates (x_i, y_i). The outputs set is generated referring to the Navier solution:

$$w(x, y) = \frac{16p}{\pi^6 \overline{D}} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{sin\frac{m\pi x}{l} sin\frac{n\pi x}{l}}{mn(\frac{m^2}{l^2} + \frac{n^2}{l^2})^2} \quad \text{for } m = 1, 3, 5, ... \text{ and } n = 1, 3, 5, ... \tag{6.24}$$

From Eq. (6.24) the number of sinusoidal necessary to approximate the displacement is infinite, but for computational reason the values of m and n are limited to 20.

Also in this case the inputs set is mapped into the dimensionless 2D domain $D_{a2} : [-1, 1] \times [-1, 1]$ through a linear transformation of each pair of coordinates in $\boldsymbol{X_L}$, such that the new inputs are:

$$\begin{cases} x_{ai} = -1 + \dfrac{x_i}{l} \\ y_{ai} = -1 + \dfrac{y_i}{l} \end{cases} \tag{6.25}$$

6.3.2. Neural network set-up

Another step for the preparation of the training process is the definition of PDE and boundary conditions residuals to be evaluated in correspondence of the collocation points gathered in two sets: $\boldsymbol{X_{CD}}$ for the PDE residual and $\boldsymbol{X_{CB}}$ for boundary conditions residuals.
From Eq. (6.21), the PDE residual is:

$$R(\boldsymbol{X_{CD}}_i, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \frac{E_N(\boldsymbol{X_{CD}}_i) t^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CD}}_i) \\ \frac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CD}}_i) \\ 2\frac{\partial^2}{\partial x \partial y} w_N(\boldsymbol{X_{CD}}_i) \end{bmatrix} - p = 0 \tag{6.26}$$

where w_N is the network approximation of the out-of-plane displacement and $E_N(x, y)$ is the Young modulus distribution approximation. In particular, $E_N(x, y)$ is expressed by means of a 2D KLE built from Eq. (3.13) and Eq. (3.14):

$$E_N(x, y) = \beta E(x, y) + \sum_{i=1}^{4} \sum_{j=1}^{4} \sqrt{\lambda_i \lambda_j} \phi_i(x) \phi_J(y) \xi_{Nij} \tag{6.27}$$

using arbitrary stochastic properties for X-direction and Y-direction.
In this case, the Young modulus approximation does not depends only on the 16 random variables ξ_{Nij}, but a new parameter β that multiplies the initial distribution $E(x, y)$ is introduced. When β is different from 1 an offset to the initial distribution is modeled. Moreover, the number of terms in Eq. (6.27) is arbitrary (theoretically it is infinite) and, for this case, it was chosen equal to sixteen for computational reasons.
It must be highlighted that, for this problem, the method works properly if at the end of the training process the parameters ξ_{Nij} converge to values close to zero, while β converge to a value such that $\beta E(x, y) \approx E(x, y) + \Delta E$.

The boundary conditions residuals are:

$$\begin{cases} B_1(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = w_N(\boldsymbol{X_{CBi}}) = 0 \\ B_2(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = \dfrac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CBi}}) = 0 \quad \text{if } \boldsymbol{X_{CBi}} = (x_{CBi}, 0) \text{ or } (x_{CBi}, l) \\ B_3(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = \dfrac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CBi}}) = 0 \quad \text{if } \boldsymbol{X_{CBi}} = (0, y_{CBi}) \text{ or } (l, y_{CBi}) \end{cases} \tag{6.28}$$

The architecture of the network is depicted in Figure 6.14.

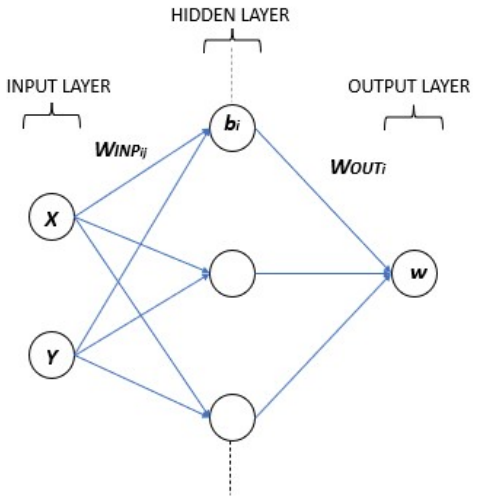


Figure 6.14: Network architecture for 2D problems

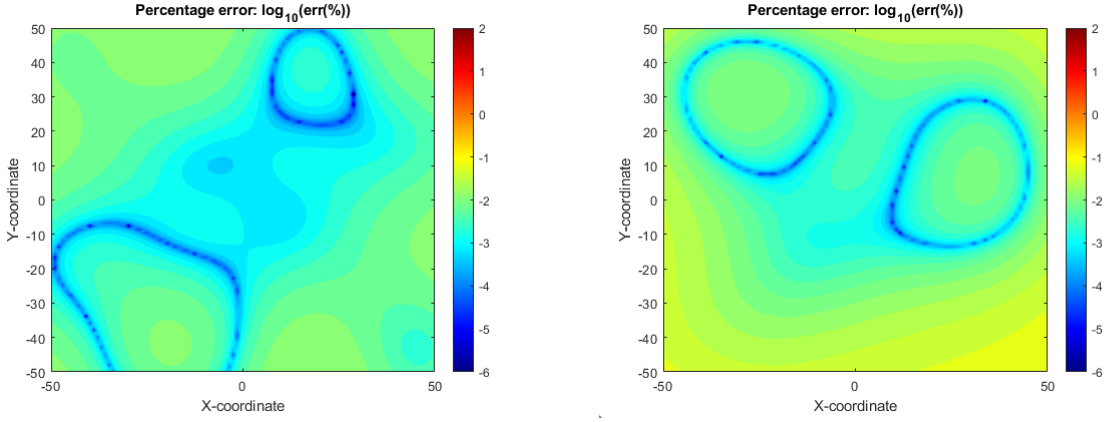
As seen, the network picks in input the in-plane spatial coordinate of a generic point of the plate (2 neurons for the input layer) and gives in output the correspondent out-of-plane displacement (1 neuron for the output layer). The number of hidden layer neurons is 3000.

The output weights, β and each ξ_{Nij} are the parameters learned by the network during the training process, for a total number of 3017 parameters. In particular, the network uses N_{CD}=400 collocation points in the domain, N_{CB}=400 collocation points along the boundaries and N_S=1500 labeled points for the training.
Moreover, the initial guess for the output weights are values between -1 and 1, while for β is 1 and for the parameters ξ_{Nij} is 0. It follows that the initial guess for $E_N(x, y)$ is $E(x, y)$.
The maximum number of iterations is fixed to 15, while the tolerance to 10^{-5}.

6.3.3. Results

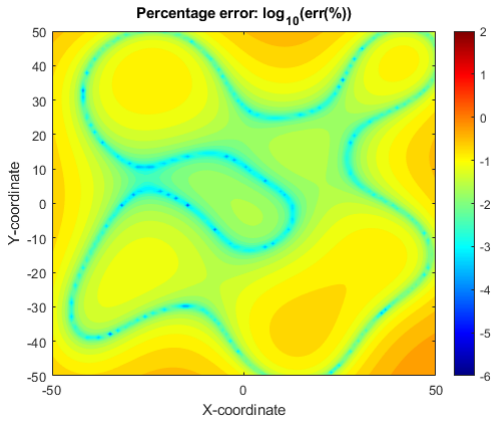
The following results are obtained for a plate of thickness t=5 mm, edge length l=100 mm and loaded by a distributed load P=10 MPa. The initial Young modulus $E(x, y)$=70000 MPa is perturbed by a constant offset $\Delta E(x, y)$=20000 MPa.

The results are depicted in Figure 6.15 and Figure 6.16. In particular, in Figure 6.15 it is reported the final percentage error distribution $err\%(x, y) = \frac{|E_N(x,y)-E_P(x,y)|}{E_P(x,y)}100$, while in Figure 6.16 is depicted the evolution of the residual vector norm. Both are represented in logarithmic scale.



(a) Percentage error using *atan*

(b) Percentage error using *tanh*

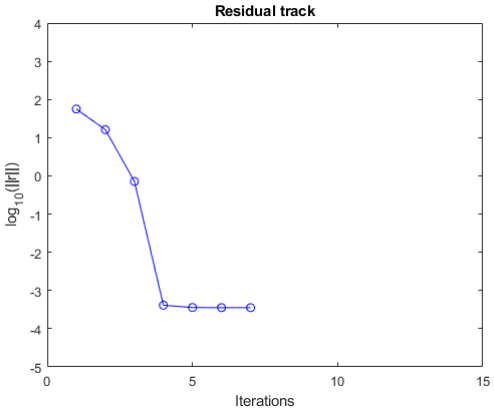


(c) Percentage error using *sigmoid*

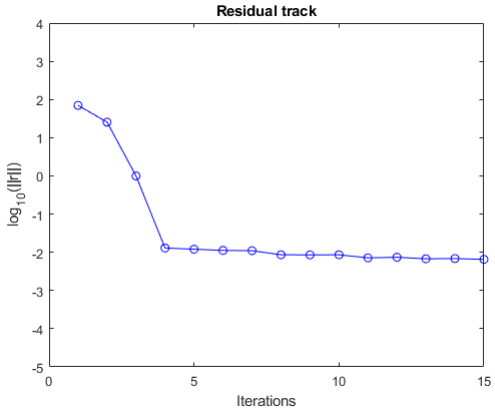
Figure 6.15: Final percentage errors distribution

Also in this case the activation function *atan* is the one that returns better result, with a percentage error in the order of 10^{-3}. *Tanh* and *sigmoid* gives worse results with percentage errors in the order of 10^{-2} and 10^{-1} respectively, but still acceptable.

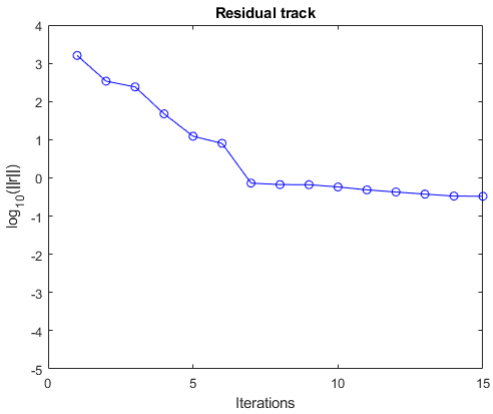
An interesting feature about the results is that, for each activation function, the maximum error is reached near the edges of the plate. This is due to the fact that, from a structural response point of view, this zone is less solicited with respect to the center of the plate. As a consequence, the second derivative of the out-of-plane displacement is really low near the edges. It follows that the PDE residual from Eq. (6.26) evaluated in proximity of the edges of the plate is low, even if a slightly large error on the elastic properties is committed.



(a) Residual vector norm evolution using *atan*



(b) Residual vector norm evolution using *tanh*



(c) Residual vector norm evolution using *sigmoid*

Figure 6.16: Residual vector norm evolution

In each case, the convergence is not reached within the maximum number of iterations fixed at the beginning of the training process. Considering that the value of the residual norm at the iteration 15 is low and that the number of iterations necessary to reach the converge depends on the value of tolerance fixed, this is not an issue.
In addition, when using *atan* the network reaches the convergence in $T_{atan} = 263.6$ seconds, while it employs $1.15T_{atan}$ seconds and $1.35T_{atan}$ seconds for *tanh* and *sigmoid*,

respectively.

The values of the internal parameters of Eq. (6.27) are summarized in Table 6.2. As seen, when using *atan* and *tanh*, the variables ξ_{Nij} converge to a values close to zero. On the contrary, the variables converge to slightly high values when *sigmoid* is adopted. This feature reflects the difference in percentage errors depicted in the Figure 6.15. However, the proposed method works properly for this kind of problem.

	ATAN	***TANH***	***SIGMOID***
β	1.2855	1.2864	1.1666
ξ_{N11}	0.0107	-0.0279	1.6835
ξ_{N12}	0.0019	-0.0092	-0.3768
ξ_{N13}	0.0075	-0.0169	-0.0057
ξ_{N14}	-0.0106	-0.0140	-.1104
ξ_{N21}	0.0013	0.0606	-0.3559
ξ_{N22}	-0.0174	-0.1485	0.0822
ξ_{N23}	-0.0200	0.0271	-0.1716
ξ_{N24}	0.0274	0.1034	0.6249
ξ_{N31}	0.0010	-0.0434	-7.7910^{-4}
ξ_{N32}	-0.0102	0.0163	0.0647
ξ_{N33}	-0.0074	-0.0379	-0.0281
ξ_{N34}	0.0203	0.0503	-0.1269
ξ_{N41}	-8.510^{-4}	0.0307	0.0370
ξ_{N42}	0.0100	0.0525	-0.1619
ξ_{N43}	0.0187	-0.0493	-0.4533
ξ_{N44}	-0.0406	0.0828	0.9465

Table 6.2: KLEs parameters at the end of the training process

6.4. Detection of fibers orientation offset for symmetric composite laminates

In the previous sections, problems regarding metallic structures are faced, but the method must be extended to the composite structures.
In particular, this section is about symmetric constant stiffness laminates. The purpose is to detect the offset of the fibers orientation with respect the nominal one for each unidirectional ply.
A square laminate with edge length l, composed by four plies with stacking sequence $[\theta_1, \theta_2, \theta_2, \theta_1]$ and thickness t_p, loaded in bending by a distributed force P and clamped along each edge, as depicted in Figure 6.17, is considered.

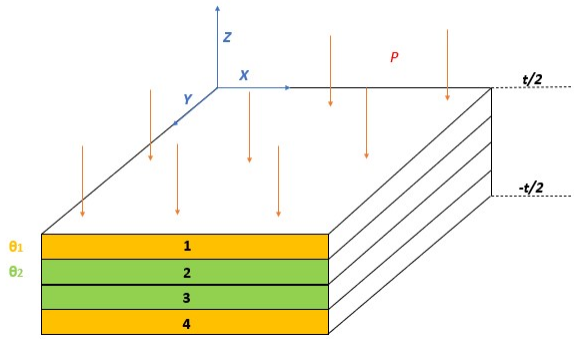


Figure 6.17: Composite constant stiffness laminate

Also in this case the Kirchhoff hypothesis are adopted, which means that the equilibrium equation along the Z-axis is the one in Eq. (6.20). Since also in this case $\hat{\boldsymbol{B}}$=0 because of the symmetry of the laminate, it can be written in matrix form as in Eq. (6.21).
The main difference with respect to the structural model reported in the previous section is that the $\hat{\boldsymbol{D}}$ matrix cannot be written as in Eq. (6.22), but it must be computed referring to the *lamination theory* reported in Section 2.1.2. Considering that the reference plane is placed between the inner plies and that the laminate thickness is $4t_p = t$, the $\hat{\boldsymbol{D}}$ matrix is:

$$\hat{\boldsymbol{D}} = \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{1}}\hat{\boldsymbol{D}}_{\boldsymbol{L1}}\hat{\boldsymbol{T}}_{\boldsymbol{1}}{}^T + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{2}}\hat{\boldsymbol{D}}_{\boldsymbol{L2}}\hat{\boldsymbol{T}}_{\boldsymbol{2}}{}^T + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{3}}\hat{\boldsymbol{D}}_{\boldsymbol{L3}}\hat{\boldsymbol{T}}_{\boldsymbol{3}}{}^T + \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{4}}\hat{\boldsymbol{D}}_{\boldsymbol{L4}}\hat{\boldsymbol{T}}_{\boldsymbol{4}}{}^T \tag{6.29}$$

where $\hat{\boldsymbol{T}}_{\boldsymbol{1}} = \hat{\boldsymbol{T}}_{\boldsymbol{4}}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{2}} = \hat{\boldsymbol{T}}_{\boldsymbol{3}}$ are the rotation matrices dependent on the fibers orientation

angle given by:

$$\hat{\boldsymbol{T}}_{\boldsymbol{i}} = \begin{bmatrix} m_i^2 & n_i^2 & m_i n_i \\ n_i 2 & m_i^2 & -m_i n_i \\ -2m_i n_i & 2m_i n_i & m_i^2 - n_i^2 \end{bmatrix} \tag{6.30}$$

in which $m_i = \cos(\theta_i)$ and $n_i = \sin(\theta_i)$.
$\hat{\boldsymbol{D}}_{\boldsymbol{L1}} = \hat{\boldsymbol{D}}_{\boldsymbol{L4}} = \hat{\boldsymbol{D}}_{\boldsymbol{L2}} = \hat{\boldsymbol{D}}_{\boldsymbol{L3}}$ are the ply stiffness matrices in the lamina reference system (see Figure 2.2) obtained from the *rule of mixtures* illustrated in Section 2.1.1. In particular:

$$\hat{\boldsymbol{D}}_{\boldsymbol{Li}} = \begin{bmatrix} \frac{E_{11i}}{1-\nu_{12i}\nu_{21i}} & \frac{E_{11i}\nu_{12i}}{1-\nu_{12i}\nu_{21i}} & 0 \\ \frac{E_{22i}}{1-\nu_{12i}\nu_{21i}} & \frac{E_{22i}\nu_{21i}}{1-\nu_{12i}\nu_{21i}} & 0 \\ 0 & 0 & G_{12i} \end{bmatrix} \tag{6.31}$$

The structural model is completed with the following boundary condition:

$$w = 0 \quad \text{for } x = 0, l \text{ and } y = 0, l \tag{6.32}$$

Starting from a nominal stacking sequence $[\theta_1, \theta_2, \theta_2, \theta_1]$, the offsets $\Delta\theta_1$ and $\Delta\theta_2$ are introduced such that the real layup is $[\theta_1 + \Delta\theta_1, \theta_2 + \Delta\theta_2, \theta_2 + \Delta\theta_2, \theta_1 + \Delta\theta_1]$.
Then, the neural network is trained to detect these offsets by using the nominal sequence as initial guess. In particular, the offsets found by the network, $\theta_{1N}(x, y)$ and $\theta_{2N}(x, y)$, are expressed by means of 2D KLEs, similarly to Eq. (6.27). This is necessary to reproduce a real situation, in which it must be assumed that the offsets are random and not constant. It follows that the during the training process the variables $\theta_{1N}(x, y)$ and $\theta_{2N}(x, y)$ has to be learned.

6.4.1. Generation of labeled inputs and outputs sets

It is necessary to generate the labeled inputs set $\boldsymbol{X}_{\boldsymbol{L}}$ and labeled outputs set $\boldsymbol{O}_{\boldsymbol{L}}$ necessary for the training process. The inputs set contains the pair of in-plane coordinates (x_i, y_i) of N_S points, while the outputs set the correspondents out-of-plane displacements.

The value of the out-of plane displacement at the labeled points is obtained through numerical analysis, exploiting a 2D finite elements model realized in ABAQUS.
The model is realized by means of 4-noded layered shell elements with 3 integration points through the thickness for each ply. A pressure load is applied to model the distributed

load P, while the *'encastre'* boundary condition is assigned to each edge.
At the end of the static analysis, the text file regarding the out-of-plane displacement is exported from ABAQUS to be given as input to a MATLAB subroutine. The latter gives in output the labeled inputs and outputs set.

Also in this case, the inputs set is mapped into the dimensionless 2D domain D_{a2} : $[-1,1]\times[-1,1]$ as reported in Eq. (6.25).

6.4.2. Neural network set-up

The residual of the PDE, which is evaluated at a set of N_{CD} points $\boldsymbol{X_{CDi}}$, is:

$$R(\boldsymbol{X_{CDi}},\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \hat{\boldsymbol{D}}_{\boldsymbol{N}}(\boldsymbol{X_{CDi}}) \begin{bmatrix} \frac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial x \partial y} w_N(\boldsymbol{X_{CDi}}) \end{bmatrix} - P = 0 \tag{6.33}$$

where $\hat{\boldsymbol{D}}_{\boldsymbol{N}}$ is the network approximation of the laminate stiffness matrix. In particular, from Eq. (6.29):

$$\hat{\boldsymbol{D}}_{\boldsymbol{N}} = \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}\hat{\boldsymbol{D}}_{\boldsymbol{L1}}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}\hat{\boldsymbol{D}}_{\boldsymbol{L2}}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}\hat{\boldsymbol{D}}_{\boldsymbol{L3}}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}^{\,T} + \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}\hat{\boldsymbol{D}}_{\boldsymbol{L4}}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}^{\,T} \tag{6.34}$$

where laminae stiffness matrices $\hat{\boldsymbol{D}}_{\boldsymbol{L1}} = \hat{\boldsymbol{D}}_{\boldsymbol{L4}} = \hat{\boldsymbol{D}}_{\boldsymbol{L2}} = \hat{\boldsymbol{D}}_{\boldsymbol{L3}}$ are the ones obtained applying the *rule of mixtures*, while the rotation matrices (see Eq. (6.30)) depend on the network approximations of the offsets. Since the laminate is symmetric, $\hat{\boldsymbol{T}}_{\boldsymbol{1N}} = \hat{\boldsymbol{T}}_{\boldsymbol{4N}}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{2N}} = \hat{\boldsymbol{T}}_{\boldsymbol{3N}}$ and, so, there are only two unknowns offsets instead of four: $\theta_{1N}(x,y)$ and $\theta_{2N}(x,y)$.
The latter are expressed by means of 2D KLEs:

$$\begin{aligned} \theta_{1N}(x,y) &= \beta_{\theta_1}\theta_1(x,y) + \sum_{i=1}^{4}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_1 ij} \\ \theta_{2N}(x,y) &= \beta_{\theta_2}\theta_2(x,y) + \sum_{i=1}^{10}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_2 ij} \end{aligned} \tag{6.35}$$

such that $\beta_{\theta_1}, \beta_{\theta_2}$, each $\xi_{\theta_1 ij}$ and each $\xi_{\theta_2 ij}$ belong to the internal parameters vector of the

network.
As for the previous case, it is expected the convergence to values close to zero for each $\xi_{\theta_1 ij}$ and $\xi_{\theta_2 ij}$, while β_{θ_1} and β_{θ_2} have to reach values such that $\beta_{\theta_1}\theta_1(x,y) \approx \theta_1(x,y)+\Delta\theta_1$ and $\beta_{\theta_2}\theta_2(x,y) \approx \theta_2(x,y) + \Delta\theta_2$.

The boundary condition residual, which is evaluated at a set of N_{CB} collocation points, is:

$$B(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = w(\boldsymbol{X_{CBi}}) = 0 \tag{6.36}$$

The architecture is the one depicted in Figure 6.14. Two neurons for the input layer, one for the output layer and 8000 for the hidden layer.
The training process has to learn 8034 internal parameters (8000 output weights, $\beta_{\theta 1}$, $\beta_{\theta 2}$, $\xi_{\theta 2ij}$ and $\xi_{\theta 2ij}$). In particular, N_S=1500 labeled points, N_{CD}=400 collocation points inside the domain and N_{CB}=400 collocation points along the boundaries are used.
Moreover, the initial guesses for the output weights are random values between -1 and 1, $\beta_{\theta 1}$ and $\beta_{\theta 2}$ are initialized to 1, while $\xi_{\theta 1xj}$ and $\xi_{\theta 2xj}$ to zero. In this way the initial guess for the stacking sequence is equal to the nominal one.
The maximum number of iterations is fixed to 15, while the tolerance to 10^{-5}.

6.4.3. Results

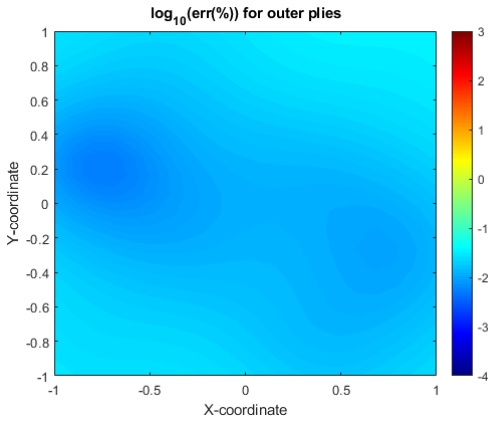
The following results are referred to a square symmetric laminate, clamped along each edge of dimension l=50 mm and loaded by a pressure load P=10 MPa. Its plies of thickness $t_p = 0.5mm$ are stacked together according to the sequence [15°,30°,30°,15°]. The latter is perturbed by means of $\Delta\theta_1$=1.3° and $\Delta\theta_2$=2.4°, such that the final sequence is [16.3°,32.4°,32.4°,16.3°].
Moreover, the elastic properties necessary to build the laminae stiffness matrices $\hat{\boldsymbol{D}}_{\boldsymbol{L1}} = \hat{\boldsymbol{D}}_{\boldsymbol{L4}} = \hat{\boldsymbol{D}}_{\boldsymbol{L2}} = \hat{\boldsymbol{D}}_{\boldsymbol{L3}}$ are: E_{11}=139000 MPa, E_{22}=10000 MPa, ν_{12}=0.34 and G_{12}=3400 MPa.

The KLEs expressing the network approximations of the offsets are built considering plausible stochastic properties in X and Y directions. In particular:

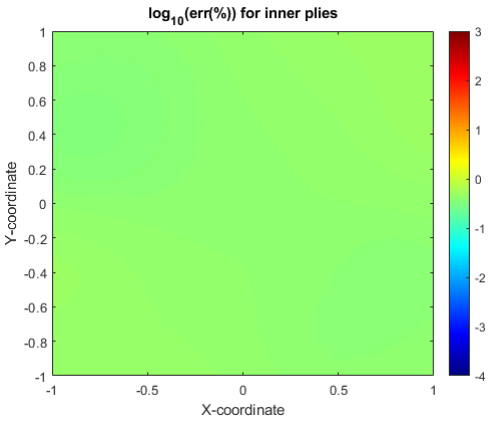
- correlation lengths $c_X = l$ and $c_Y = l$;
- Exponential correlation function $C(x_1, y_1, x_2, y_2) = e^{\frac{-|x_1-x_2|}{c_X}} e^{\frac{-|y_1-y_2|}{c_Y}}$;

- Variances σ_X^2=5° and σ_Y^2=5°;

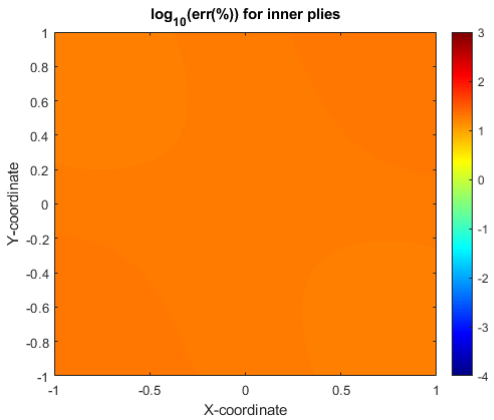
The training process is performed for each hidden layer activation function. When *sigmoid* is adopted the results are not good at all and, so, only the ones obtained using *atan* and *tanh* are reported in Figure 6.18.
In particular, in Figure 6.18 there are depicted the final percentage error distributions, $err_{\theta 1}\%(x,y) = \frac{|\theta_{1N}(x,y)-(\theta_1+\Delta\theta_1)|}{(\theta_1+\Delta\theta_1)}$ and $err_{\theta 2}\%(x,y) = \frac{|\theta_{2N}(x,y)-(\theta_2+\Delta\theta_2)|}{(\theta_2+\Delta\theta_2)}$, regarding the fibers orientation distributions for the inner plies and outer plies, respectively. Both are reported in logarithmic scale.



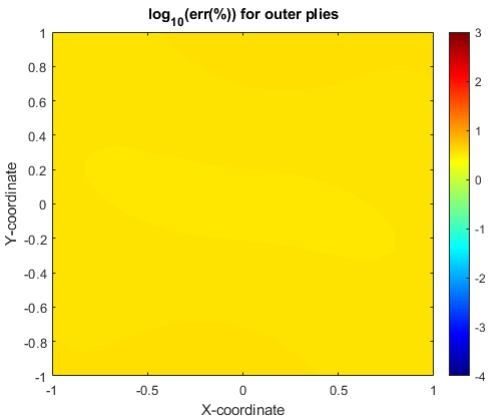
(a) Percentage error on outer plies using *atan*



(b) Percentage error on inner plies using *atan*



(c) Percentage error on outer plies using *tanh*



(d) Percentage error on inner plies using *sigmoid*

Figure 6.18: Final percentage errors distributions

It is possible to notice that percentage errors are really low when *atan* is adopted, in the order of 10^{-2} for the outer plies and 10^{-1} for the inner plies. For *tanh*, instead, the error is still acceptable for the outer plies, but it becomes slightly high for the inner plies reaching the order of 10^1.

Interesting features about the error are that the maximum values are reached always near the edges of the plies and that the inner plies error is always higher than the one referred to the outer plies.
As for the previous problem, the first feature is due to the fact that the loading condition adopted does not solicit the laminate near the edges as it does at the center.
The second feature, instead, is due to the fact that the inner plies contribution to the structural response is less important with respect to the one given by the outer plies (see Eq. (6.34)). It follows that an error on the elastic properties for the inner plies has a minor effect on the PDE residual (see Eq. (6.33)) with respect to the same error committed for the outer plies.

In order to understand why the adoption of *sigmoid* leads to bad results, it is interesting to analyze the behaviour of the residual vector norm in Figure 6.19.

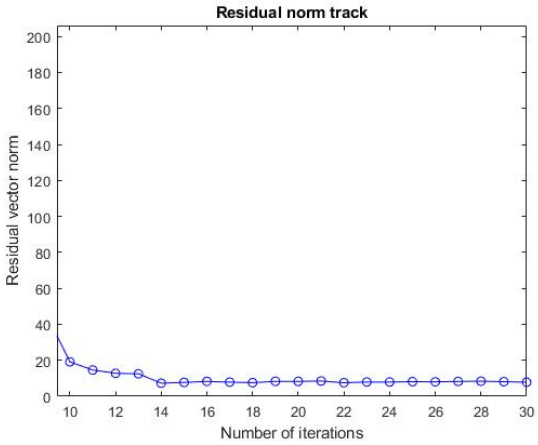
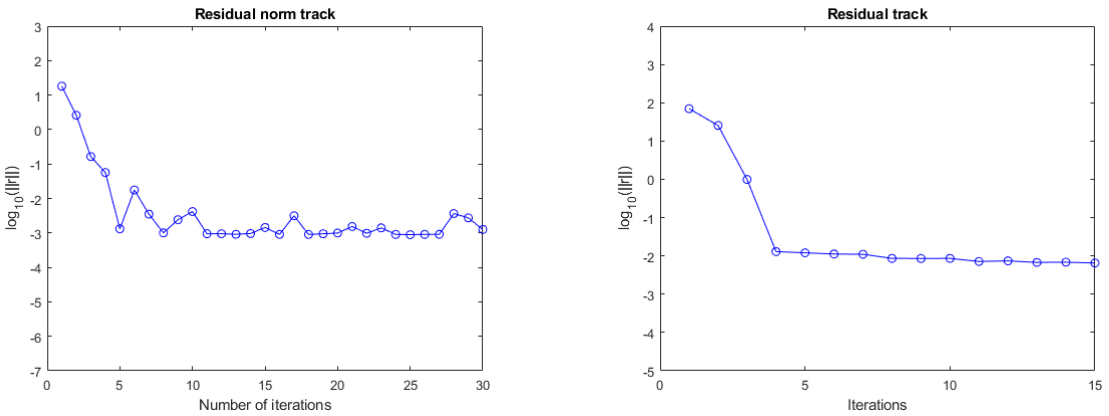


Figure 6.19: Residual vector norm evolution when using *sigmoid*

In particular, the evolution from iteration 10 up to 30 is reported. As seen, the residual vector norm does not seem to decrease, but it remains stucked to a value around 7. This can be addressed to a presence of a local minimum (and it can be solved by changing the initial guesses) or to the fact that *sigmoid* is not a good choice as activation function.

For the sake of completeness, the evolution of the residual vector norm for the other two cases are reported in Figure 6.20 .



(a) Residual vector norm evolution using *atan* (b) Residual vector norm using *tanh*

Figure 6.20: Residual vector norm evolution

When *atan* and *tanh* are adopted, the network does not reach the convergence before the iteration 15. The main difference with respect to the *sigmoid* case is that the value of the residual norm is in the order of 10^{-3} after few iterations and, so, the convergence is expected. Moreover, the number of iterations necessary to reach the convergence is strictly dependent on the value of the tolerance fixed at the beginning of the training process.

6.5. Detection of plies fiber misalignment for symmetric VAT laminates

The last step is the application of the proposed method for the detection of defects in VAT structures. In this section, the goal is to find the fibers misalignment for each ply of a symmetric VAT laminate. In particular, the latter is loaded in bending by a distributed load P and clamped along each edge of length l, as depicted in Figure 6.21.

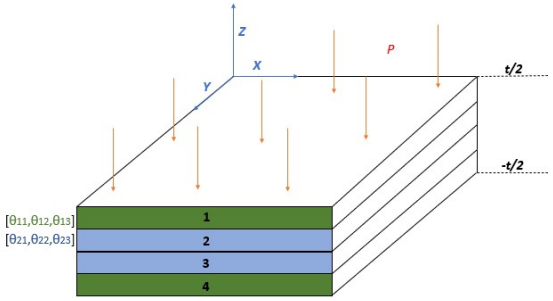


Figure 6.21: VAT laminate

The laminate is made of four plies of thickness t_p, whose nominal fibers paths follow a non-linear law as the one in Eq. (2.9). Since a quadratic law is considered, the fibers path is defined by a vector $[\theta_1, \theta_2, \theta_3]$ containing the fibers orientation at the reference points. As example, the quadratic path obtained considering $A = (0, \frac{l}{2})$, $B = (\frac{l}{2}, \frac{l}{2})$ and $C = (l, \frac{l}{2})$ as reference points is reported in Figure 6.22.

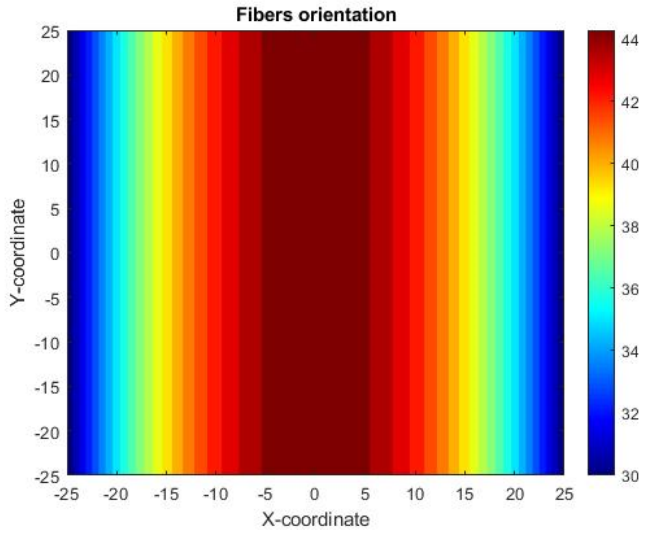


Figure 6.22: Example of quadratic variation for [30°,45°,30°]

Starting from two nominal quadratic fibers paths, $\theta_1(x, y)$ and $\theta_2(x, y)$, they are perturbed by means of two different known KLEs and they are assigned to the laminate in a symmetric way. Then, the neural network is trained to learn the perturbed distributions using as initial guesses the nominal ones.

6.5.1. Generation of labeled inputs and outputs sets

For the generation of the labeled inputs set $\boldsymbol{X_L}$ and outputs set $\boldsymbol{O_L}$, a finite elements model is strictly necessary. The input file for the ABAQUS software is generated with a MATLAB subroutine. Firstly, it defines the nodes and their connectivity for the creation of the elements. Then, the model grid is superposed on the fibers path of each ply (see Figure 6.23), in order to find the values of the angles at the centroid of each element and to define its composite section. It follows that the number of composite section of the model is equal to the number of elements.

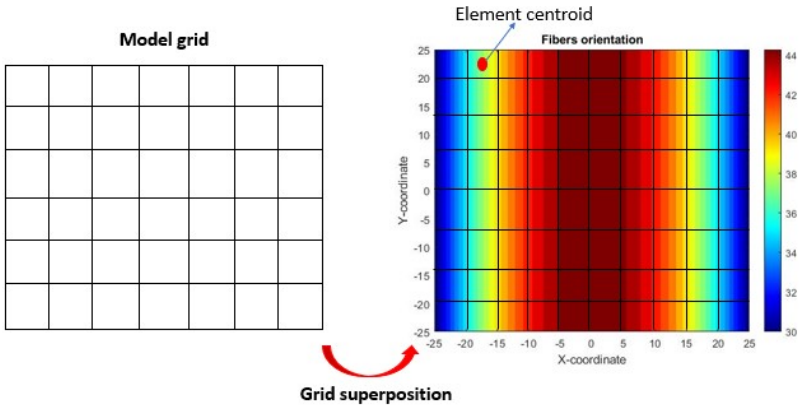


Figure 6.23: Element section definition

Moreover, the material elastic properties, the boundary conditions for the edges and the pressure load P are defined.
The input file returns a numerical model made of shell elements, whose number is tuned assuming that the variation of fibers angle between adjacent elements does not have to be too wide. It is obvious that the results from the numerical analysis would be more reliable if more elements are considered. An analysis of converge led to use 2500 elements for this model.

After the static analysis, a text file containing the out-of plane displacements is exported from ABAQUS to be given as input to another MATLAB subroutine. The latter gives in output the desired labeled sets. Then, the inputs set is mapped onto the dimensionless domain as in Eq. (6.25).

6.5.2. Neural network set-up

The residual of the equilibrium equation, which is evaluated at a set of N_{CD} points $\boldsymbol{X_{CDi}}$, is:

$$R(\boldsymbol{X_{CDi}}, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \hat{\boldsymbol{D}}_{\boldsymbol{N}}(\boldsymbol{X_{CDi}}) \begin{bmatrix} \frac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial x \partial y} w_N(\boldsymbol{X_{CDi}}) \end{bmatrix} - P = 0 \tag{6.37}$$

where $\hat{\boldsymbol{D}}_{\boldsymbol{N}}$ is the network approximation of the laminate stiffness matrix. In particular, from Eq. (6.29):

$$\hat{\boldsymbol{D}}_{\boldsymbol{N}} = \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}\hat{\boldsymbol{D}}_{\boldsymbol{L1}}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}^{T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}\hat{\boldsymbol{D}}_{\boldsymbol{L2}}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}^{T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}\hat{\boldsymbol{D}}_{\boldsymbol{L3}}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}^{T} + \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}\hat{\boldsymbol{D}}_{\boldsymbol{L4}}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}^{T} \tag{6.38}$$

where the laminae stiffness matrices are the ones obtained from the *rule of mixtures*, while the rotation matrices (see Eq. (6.30)) depend on the network approximations of the fibers paths. Because of the symmetry of the laminate, it can be assumed that $\hat{\boldsymbol{T}}_{\boldsymbol{1N}} = \hat{\boldsymbol{T}}_{\boldsymbol{4N}}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{2N}} = \hat{\boldsymbol{T}}_{\boldsymbol{3N}}$. It follows that there are only two unknowns paths to be learned: $\theta_{1N}(x, y)$ and $\theta_{2N}(x, y)$.
In particular, the latter are expressed by means of 2D KLEs:

$$\begin{aligned}
\theta_{1N}(x,y) &= \theta_1(x,y) + \sum_{i=1}^{4}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_1 ij} \\
\theta_{2N}(x,y) &= \theta_2(x,y) + \sum_{i=1}^{10}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_2 ij}
\end{aligned} \tag{6.39}$$

where $\theta_1(x, y)$ and $\theta_2(x, y)$ are not constant, but their expressions reflect Eq. (2.9). Since a quadratic rule is adopted for the nominal paths and the reference points considered are $A = (0, \frac{l}{2}), B = (\frac{l}{2}, \frac{l}{2})$ and $C = (l, \frac{l}{2})$, it follows that:

$$\begin{aligned}
\theta_1(x,y) &= \theta_{1A}\frac{(x-\frac{l}{2})}{-\frac{l}{2}}\frac{(x-l)}{-l} + \theta_{1B}\frac{x}{\frac{l}{2}}\frac{(x-l)}{-\frac{l}{2}} + \theta_{1C}\frac{x}{-l}\frac{(x-\frac{l}{2})}{\frac{l}{2}} \\
\theta_2(x,y) &= \theta_{2A}\frac{(x-\frac{l}{2})}{-\frac{l}{2}}\frac{(x-l)}{-l} + \theta_{2B}\frac{x}{\frac{l}{2}}\frac{(x-l)}{-\frac{l}{2}} + \theta_{2C}\frac{x}{-l}\frac{(x-\frac{l}{2})}{\frac{l}{2}}
\end{aligned} \tag{6.40}$$

where θ_{1A}, θ_{1B} and θ_{1C} are the values of the fibers path at the reference points (A,B,C) for the outer plies, while θ_{2A}, θ_{2B} and θ_{2C} for the inner plies. The parameters $\xi_{\theta_1 ij}$ and $\xi_{\theta_2 ij}$ belong to the internal parameters vector of the network.
Moreover, the stochastic properties considered to build the expansions in Eq. (6.39) are the same used to define the KLEs for the creation of the perturbed paths $\theta_{1P}(x, y)$ and $\theta_{2P}(x, y)$.

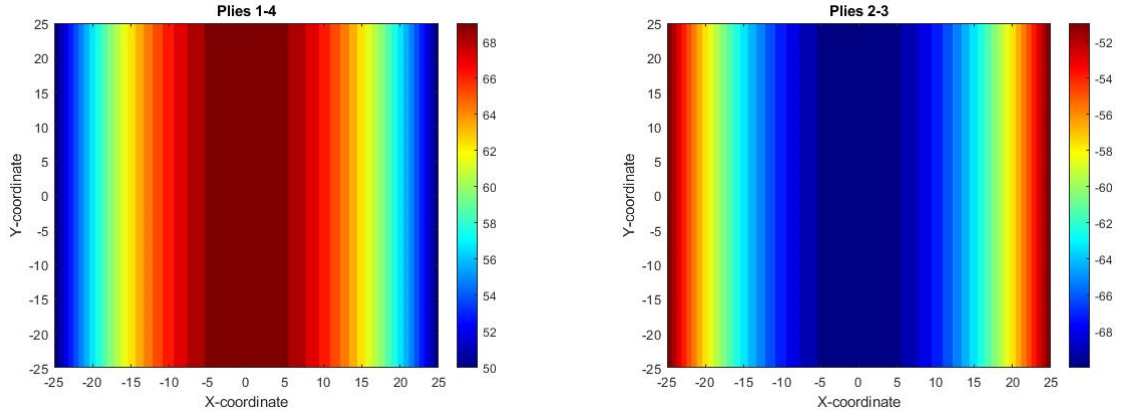
Regarding the boundary condition residual, it is evaluated at a set of N_{CB} collocation points and it is:

$$B(\boldsymbol{X}_{\boldsymbol{CBi}}, \boldsymbol{\theta}) = w(\boldsymbol{X}_{\boldsymbol{CBi}}) = 0 \tag{6.41}$$

The network architecture is the one depicted in Figure 6.14. So, 2 neurons for the input layer, 1 for the output layer and 12000 for the hidden layer.
The training process has to learn 12032 internal parameters (12000 output weights, $\xi_{\theta 2ij}$ and $\xi_{\theta 2ij}$). In particular, N_S=2500 labeled points, N_{CD}=400 collocation points inside the domain and N_{CB}=400 collocation points along the boundaries are used.
Moreover, the initial guesses for the output weights are values between -1 and 1, while each unknown parameter in Eq. (6.39) is set to 0. It follows that initial guess for $\theta_{1N}(x, y)$ is $\theta_1(x, y)$, while for $\theta_{2N}(x, y)$ is $\theta_2(x, y)$.
The maximum number of iterations is fixed to 15, while the tolerance to 10^{-5}.

6.5.3. Results

The following results are referred to a laminate made of four plies with thickness $t_p = 0.2$, edge length $l = 50$ mm and loaded by a distributed force $P = 0.5$ MPa. Moreover, the composite material has following elastic properties: $E_{11} = 139000$ MPa, $E_{22} = 10000$ MPa, $\nu_{12} = 0.34$ and $G_{12} = 3000$ MPa.
The nominal fibers paths follow a quadratic rule characterized by $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$=[50°,70°,50°] and $[\theta_{2A}, \theta_{2B}, \theta_{2C}]$=[-50°,-70°,-50°], as depicted in Figure 6.24.



(a) Nominal fibers distribution for outer plies (b) Nominal fibers distribution for inner plies

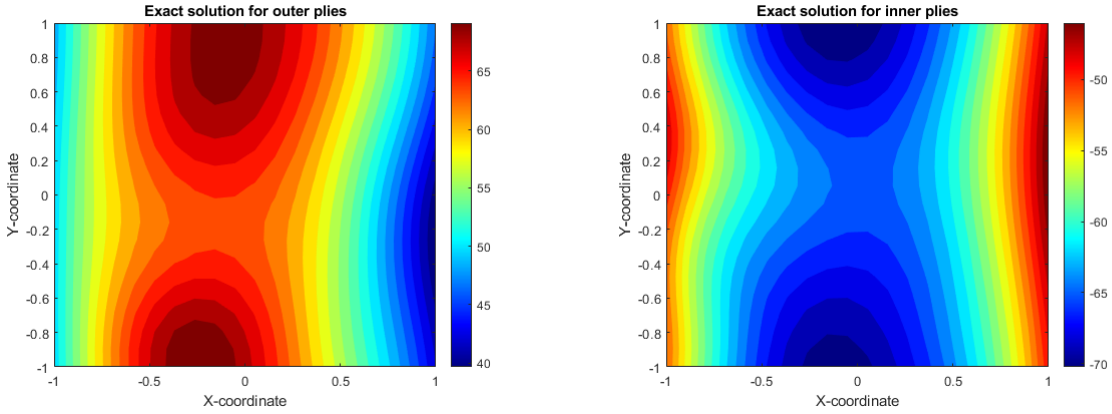
Figure 6.24: Fibers nominal distributions

The latter are perturbed considering the following stochastic properties:

- correlation lengths $c_X = l$ and $c_Y = l$;

- Exponential correlation function $C(x_1, y_1, x_2, y_2) = e^{\frac{-|x_1-x_2|}{c_X}} e^{\frac{-|y_1-y_2|}{c_Y}}$;
- Variances σ_X^2=5° and σ_Y^2=5°;

The perturbed fibers paths assigned to the numerical model are depicted in Figure 6.25. Moreover, they are the expected fibers distributions to be found from the training process.

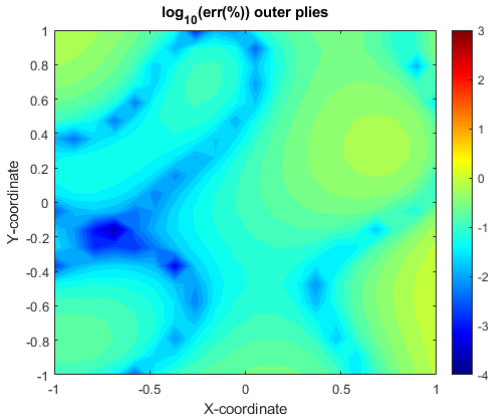


(a) Fibers distribution for outer plies

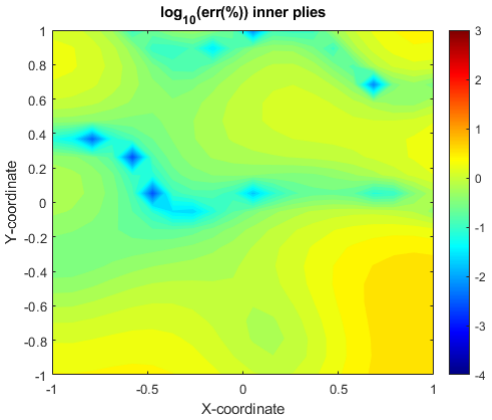
(b) Fibers distribution for inner plies

Figure 6.25: Fibers distributions

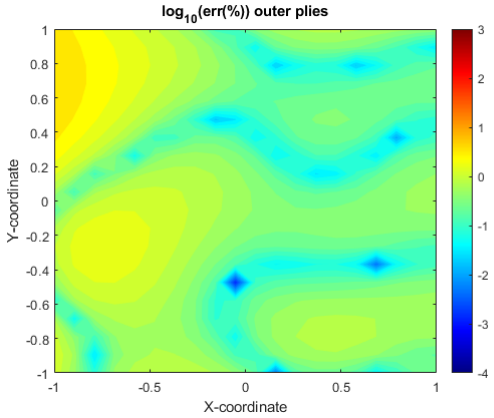
Adopting *sigmoid* as hidden layer activation function leads to very bad results. In Figure 6.26 are depicted the ones obtained using *atan* and *tanh*. In particular, there are reported the final percentage error distributions, $err_{14}\% = \frac{|\theta_{1N}(x,y)-\theta_{1P}(x,y)|}{\theta_{1P}(x,y)}$, $err_{23}\% = \frac{|\theta_{2N}(x,y)-\theta_{2P}(x,y)|}{\theta_{2P}(x,y)}$, for the outer plies and inner plies, respectively. Both are reported in logarithmic scale.



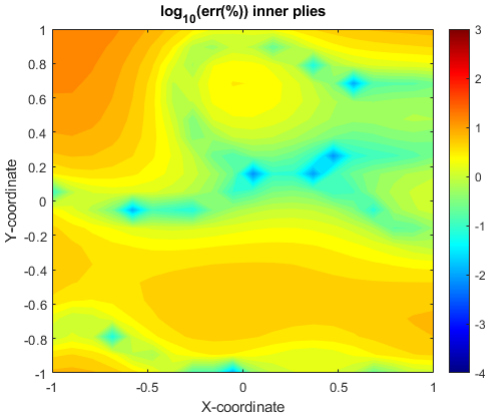
(a) Percentage error for outer plies using textitatan



(b) Percentage error for inner plies using *atan*



(c) Percentage error for outer plies using *tanh*



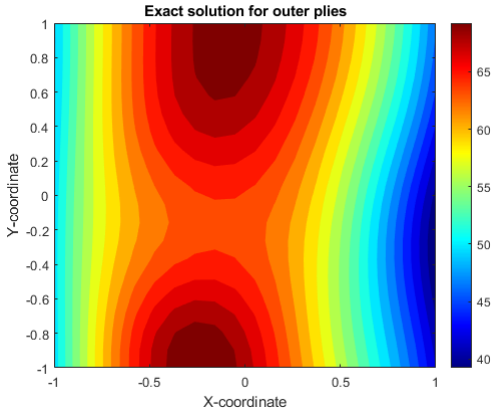
(d) Percentage error for inner plies using *tanh*

Figure 6.26: Final percentage error distributions

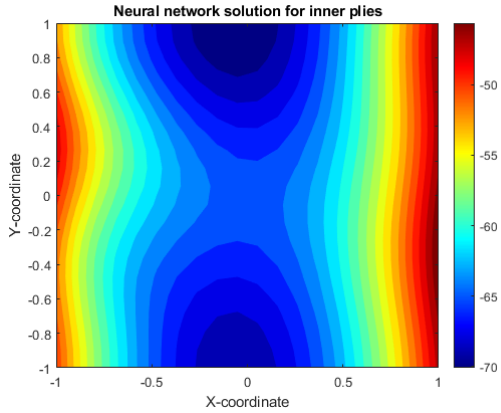
As seen, *atan* is the activation function that leads to the best results, with a percentage error in the order of 10^{-1} for the outer plies and 10^{0} for the inner plies. The situation gets worse when *tanh* is used, with percentage errors that are almost one order higher than the previous ones.

Moreover, the percentage errors are higher for the inner plies than the outer plies and their maximum values are always reached near the edges of the laminate. The reasons for this behaviour were explained in the previous sections.

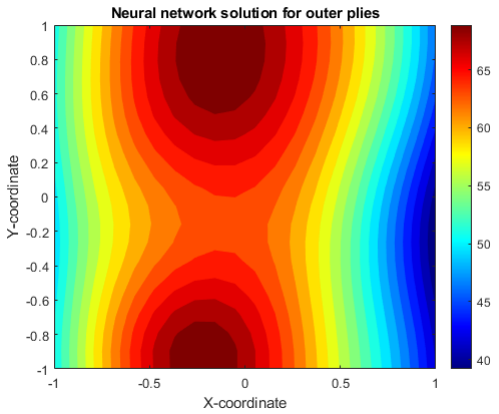
In order to make a direct comparison with Figure 6.25, in Figure.6.26 are depicted $\theta_{1N}(x, y)$ and $\theta_{2N}(x, y)$.



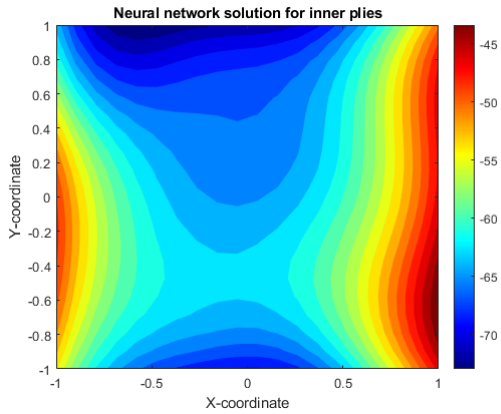
(a) Network solution for outer plies using *atan*



(b) Network solution for inner plies using *atan*



(c) Network solution for outer plies using *tanh*

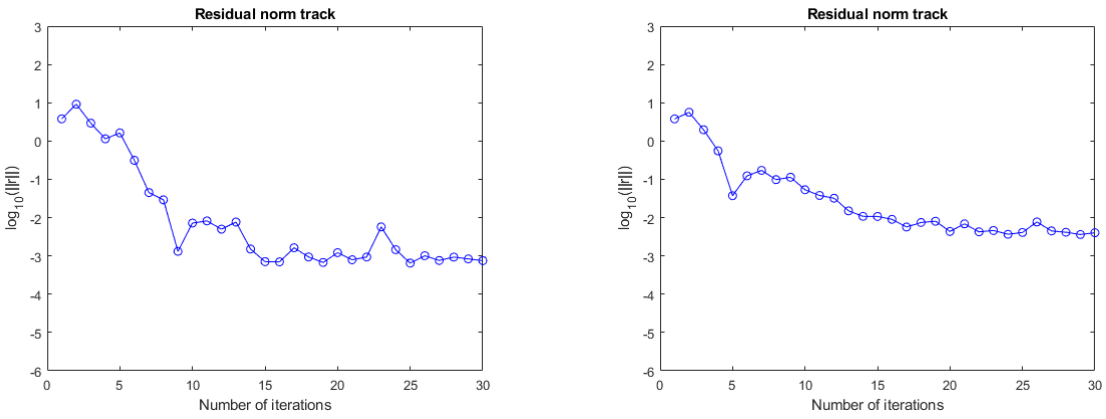


(d) Network solution for inner plies using *tanh*

Figure 6.27: Network solutions

Regarding the *sigmoid*, the same issue of the previous problem is encountered. As a matter of fact, the value of the residual vector norm at the iteration 15 is still too high. Moreover, even if the maximum number of iterations is fixed to 30, the trend does not suggest a possible convergence because the value of the residual vector norm remains high and almost the same between the iterations. This can be addressed to a presence of a local minimum or to the fact that *sigmoid* is not a good choice for this type of problem.

As depicted in Figure 6.28, the training process does not reach the convergence within the fixed maximum number of iterations neither when *atan* and *tanh* are adopted. However, the value of the residual vector norm at the iteration 15 is low (in the order of 10^{-2}) and, so, the convergence is plausible. Moreover, the number of iterations to reach the convergence is strictly dependent on the value of tolerance fixed at the beginning.



(a) Residual vector norm evolution using *atan* (b) Residual vector norm using *tanh*

Figure 6.28: Residual vector norm evolution

6.6. Detection of gaps and overlaps for VAT laminates

A common defect affecting VAT structure is the presence of gaps and overlaps. In this section, the effectiveness of the method in detecting these defects is tested. In particular, it is assumed that their presence can be considered as a variation of the fibers volumetric fraction of the laminate with respect to the nominal value.

It is considered a symmetric laminate (see Figure 6.21) characterized by a constant fibers volumetric fraction V_f and nominal fibers paths, $\theta_1(x, y)$ and $\theta_2(x, y)$, for its outer and inner plies, respectively. Then, the volumetric fraction is perturbed by means of a 2D KLE built considering plausible stochastic properties, such that the real volumetric fraction becomes $V_{fP}(x, y)$. The fibers path, instead, are not perturbed.
The goal of the method is to reconstruct the perturbed volumetric fraction distribution and, at the same time, finds the vectors $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$ and $[\theta_{2A}.\theta_{2B}, \theta_{2C}]$, which contain the values of fibers angles at the reference points.

The generation of the labeled sets, $\boldsymbol{X_L}$ and $\boldsymbol{O_L}$, requires to perform numerical analyses in ABAQUS. The steps followed for its realization are the same reported in Section 6.5.1. The only difference is that the material elastic properties are not constant, since the volumetric fraction varies. So, for each element the value of the volumetric fraction at its centroid is extracted and used to modify the elastic properties assigned to its composite

section.

6.6.1. Neural network set-up

The residual of the PDE, which is evaluated at a set of N_{CD} points $\boldsymbol{X_{CDi}}$, is:

$$R(\boldsymbol{X_{CDi}}, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \hat{\boldsymbol{D}}_{\boldsymbol{N}}(\boldsymbol{X_{CDi}}) \begin{bmatrix} \frac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial x \partial y} w_N(\boldsymbol{X_{CDi}}) \end{bmatrix} - P = 0 \tag{6.42}$$

where $\hat{\boldsymbol{D}}_{\boldsymbol{N}}$ is the network approximation of the laminate stiffness matrix. In particular, from Eq. (6.29):

$$\hat{\boldsymbol{D}}_{\boldsymbol{N}} = \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}\hat{\boldsymbol{D}}_{\boldsymbol{L1N}}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}\hat{\boldsymbol{D}}_{\boldsymbol{L2N}}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}\hat{\boldsymbol{D}}_{\boldsymbol{L3N}}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}^{\,T} + \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}\hat{\boldsymbol{D}}_{\boldsymbol{L4N}}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}^{\,T} \tag{6.43}$$

where the rotation matrices, $\hat{\boldsymbol{T}}_{\boldsymbol{1N}} = \hat{\boldsymbol{T}}_{\boldsymbol{4N}}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{2N}} = \hat{\boldsymbol{T}}_{\boldsymbol{3N}}$, depend on the network approximation of the fibers angles at the reference points $[\theta_{1AN}, \theta_{1BN}, \theta_{1CN}]$ and $[\theta_{2AN}, \theta_{2BN}, \theta_{2CN}]$, respectively (see Eq. (6.40)). The stiffness matrices $\hat{\boldsymbol{D}}_{\boldsymbol{L1N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L2N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L3N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L4N}}$ depend on the network approximations of the fibers volumetric fraction $V_{fN}(x, y)$ (see Section 2.1.1). It follows that the elastic properties of composite material can be written as:

$$\begin{cases} E_{11N}(x,y) = E_m + (E_f - E_m)V_{fN}(x,y), \\ E_{22N}(x,y) = \dfrac{E_m E_f}{E_f + (E_m - E_f)V_{fN}(x,y)}, \\ \nu_{12N}(x,y) = \nu_m + (\nu_f - \nu_m)V_{fN}(x,y), \\ G_{12N}(x,y) = \dfrac{G_m G_f}{G_f + (G_m - G_f)V_{fN}(x,y)}. \end{cases} \tag{6.44}$$

The approximation of the fibers volumetric fraction $V_{fN}(x, y)$ is expressed by means of a 2D KLE, which is built using the same stochastic properties used to perturb $V_f(x, y)$. It follows that:

$$V_{fN}(x,y) = V_f + \sum_{i=1}^{4}\sum_{j=1}^{4} \sqrt{\lambda_i \lambda_j}\phi_i(x)\phi_j(y)\xi_{V_f ij} \tag{6.45}$$

The components of the vectors $[\theta_{1AN}, \theta_{1BN}, \theta_{1CN}]$ and $[\theta_{2AN}, \theta_{2BN}, \theta_{2CN}]$, instead, are scalar parameters that, together with the variables $\xi_{V_f ij}$, belong to the internal parameters.

The boundary condition residual, which is evaluated at a set of N_{CB} collocation points, is:

$$B(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = w(\boldsymbol{X_{CBi}}) = 0 \tag{6.46}$$

The network architecture is the one depicted in Figure 6.14. So, 2 neurons for the input layer, 1 for the output layer and 12000 for the hidden layer.
The training process has to learn 12022 internal parameters (12000 output weights, sixteen $\xi_{\theta 2ij}$, $\theta_{1AN}, \theta_{1BN}, \theta_{1CN}, \theta_{2AN}, \theta_{2BN}$ and θ_{2CN}). In particular, N_S=2500 labeled points, N_{CD}=400 collocation points inside the domain and N_{CB}=400 collocation points along the boundaries are used.
The initial guesses for the output weights are values between -1 and 1. The variables in Eq. (6.45) are initialized to 0, such that the initial guess for $V_{fN}(x,y)$ is V_f. Moreover, the initial guess for the vector $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$ is [48°,72°,51°], while for $[\theta_{2A}, \theta_{2B}, \theta_{2C}]$ is [-48°,-72°,-51°].
The maximum number of iterations is fixed to 30, while the tolerance to 10^{-5}.

6.6.2. Results

The following results are referred to a laminate made of four plies of thickness $t_p = 0.2$ mm, edge length $l = 50$ mm and loaded by a distributed force $P = 0.5$ MPa. It is made of a composite material with the following elastic properties:

Elastic property	Tensile modulus (MPa)	Poisson ratio	Shear modulus (MPa)
Fibers elastic properties	240000	0.3	92000
Matrix elastic properties	3000	0.34	1120

Table 6.3: Composite elastic properties

The fibers paths $\theta_1(x, y)$ and $\theta_2(x, y)$ follow a quadratic law characterized by $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$=[50°,70°,50 and $[\theta_{2A}, \theta_{2B}, \theta_{2C}]$=[-50°,-70°,-50°] (see Figure 6.24).
The volumetric fraction nominal value is 0.8 and it is perturbed considering the following stochastic properties:

- correlation lengths $c_X = l$ and $c_Y = l$;
- Exponential correlation function $C(x_1, y_1, x_2, y_2) = e^{\frac{-|x_1-x_2|}{c_X}} e^{\frac{-|y_1-y_2|}{c_Y}}$;
- Variances σ_X^2=0.12 and σ_Y^2=0.12;

It follows that the perturbed distribution $V_{FP}(x, y)$ used in the numerical model is depicted in Figure 6.29.

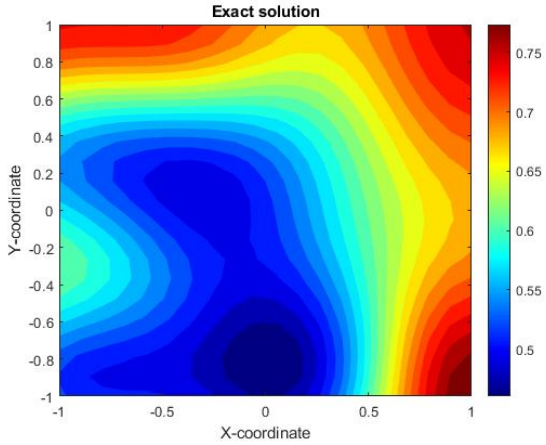


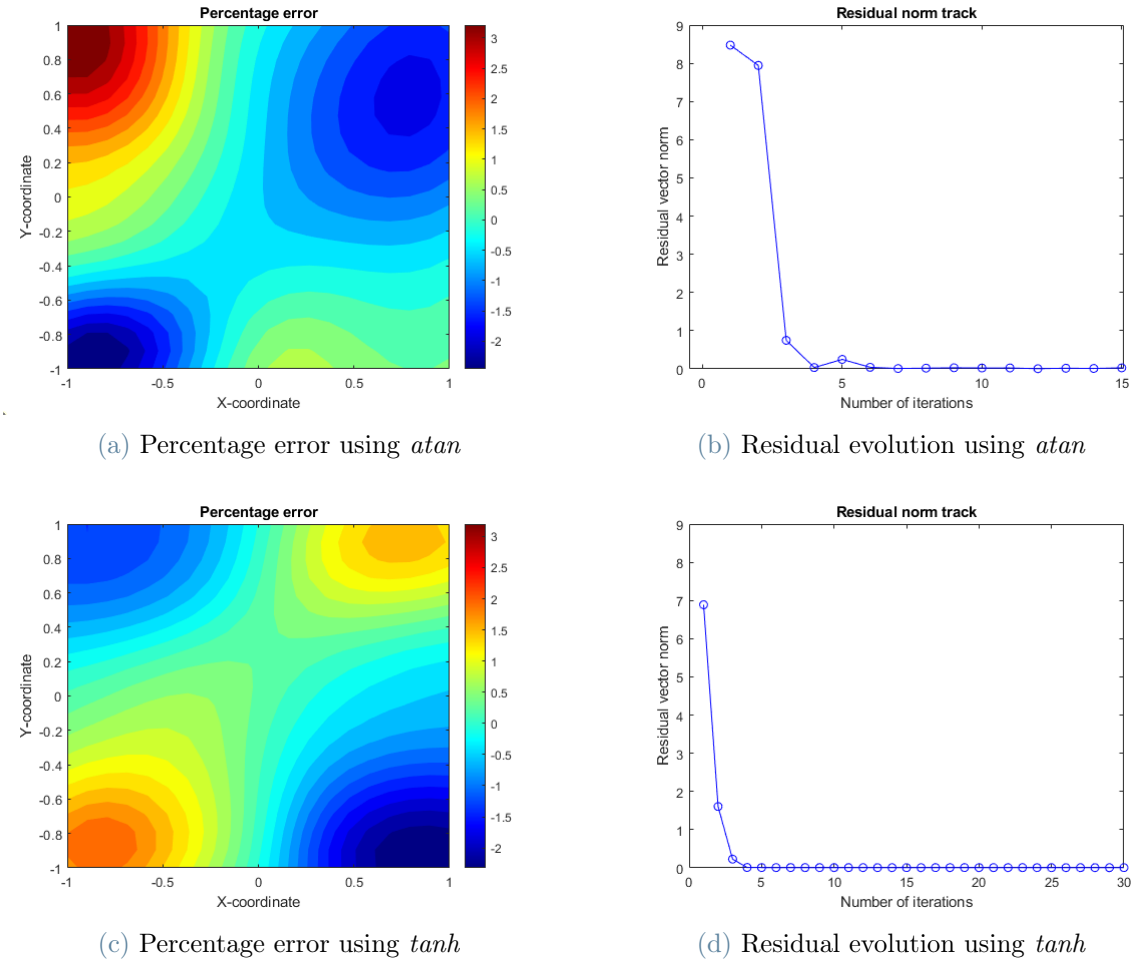
Figure 6.29: Perturbed volumetric fraction

The adoption of *sigmoid* as hidden layer activation function leads to very bad results. For this reason, only the results obtained when using *atan* and *tanh* are reported.
The values of the fibers angles at the reference points learned by the network are summarized in Table 6.4.

	θ_{1A}	θ_{1B}	θ_{1C}	θ_{2A}	θ_{2B}	θ_{2C}
ATAN	50.32°	70.32°	49.39°	-49.73°	-68.60°	-49.43°
TANH	50.26°	69.57°	49.73°	-50.11°	-71°	-50.17°
Percentage error(ATAN)	0.64%	0.46%	1.22%	0.54%	2%	1.14%
Percentage error(TANH)	0.52%	0.61%	0.54%	0.22%	1.43%	0.34%

Table 6.4: Neural network solution at the reference points

The final percentage error distribution $err_{Vf}\% = \frac{V_{fN}(x,y) - V_{fP}(x,y)}{V_{fP}(x,y)}$ and the evolution of the residual vector norm are depicted in Figure 6.30.



(a) Percentage error using *atan*

(b) Residual evolution using *atan*

(c) Percentage error using *tanh*

(d) Residual evolution using *tanh*

Figure 6.30: Final percentage errors and residual vector norm evolution

As seen, *tanh* is the activation function that leads to the best results. As a matter of fact, both percentage errors on fibers angles and volumetric fraction are slightly lower than the

ones obtained when *atan* is adopted. Moreover, the maximum error on the volumetric fraction is reached near the edges of the laminate, for the same reasons explained in Section 6.4 and Section 6.5.
As opposed to the previous cases, the percentage errors on fibers angles (see Table 6.4) for the inner plies are not always higher than the ones referred to the outer plies.
Regarding the evolution of the residual vector norm, the convergence is not reached within the maximum number of iterations. This is not an issue, since the value of the residual norm at the last iteration is low (in the order of 10^{-3}). Moreover, the number of iterations necessary to reach the convergence is strictly dependent on the value of tolerance. When *sigmoid* is adopted, instead, the residual vector norm at the iteration 30 is still high and its trend does not suggest the convergence.

6.7. Detection of fibers misalignment, gaps and overlaps for VAT laminates

The last step of this chapter is to prove the effectiveness of the method in detecting the fibers misalignment and the presence of gaps and overlaps at the same time.
It is considered a symmetric VAT laminate (see Figure 6.21)) characterized by a constant fibers volumetric fraction V_f, a quadratic fibers path $\theta_1(x, y)$ for the outer plies and a quadratic fibers path $\theta_2(x, y)$ for the inner plies.
The initial fibers volumetric fraction and fibers paths are perturbed by means of a known 2D KLEs, in order to generate a laminate with fibers misalignment, gaps and overlaps at known locations.
The method has to reconstruct both the perturbed volumetric fraction $V_{fP}(x, y)$ and the perturbed fibers paths $\theta_{1P}(x, y), \theta_{2P}(x, y)$ through the training process of the network.

The generation of the labeled inputs set $\boldsymbol{X_L}$ and outputs set $\boldsymbol{O_L}$ is done through numerical analyses performed in ABAQUS. The MATLAB subroutine illustrated in Section 6.6 is used.

6.7.1. Neural network set-up

The residual of the PDE, which is evaluated at a set of N_{CD} points $\boldsymbol{X_{CDi}}$, is:

$$R(\boldsymbol{X_{CDi}}, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial y^2} & 2\frac{\partial^2}{\partial x \partial y} \end{bmatrix} \hat{\boldsymbol{D}}_{\boldsymbol{N}}(\boldsymbol{X_{CDi}}) \begin{bmatrix} \frac{\partial^2}{\partial x^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial y^2} w_N(\boldsymbol{X_{CDi}}) \\ \frac{\partial^2}{\partial x \partial y} w_N(\boldsymbol{X_{CDi}}) \end{bmatrix} - P = 0 \tag{6.47}$$

where $\hat{\boldsymbol{D}}_{\boldsymbol{N}}$ is the network approximation of the laminate stiffness matrix. In particular, from Eq. (6.29):

$$\hat{\boldsymbol{D}}_{\boldsymbol{N}} = \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}\hat{\boldsymbol{D}}_{\boldsymbol{L1N}}\hat{\boldsymbol{T}}_{\boldsymbol{1N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}\hat{\boldsymbol{D}}_{\boldsymbol{L2N}}\hat{\boldsymbol{T}}_{\boldsymbol{2N}}^{\,T} + \frac{t^3}{24}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}\hat{\boldsymbol{D}}_{\boldsymbol{L3N}}\hat{\boldsymbol{T}}_{\boldsymbol{3N}}^{\,T} + \frac{7t^3}{192}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}\hat{\boldsymbol{D}}_{\boldsymbol{L4N}}\hat{\boldsymbol{T}}_{\boldsymbol{4N}}^{\,T} \tag{6.48}$$

where the rotation matrices, $\hat{\boldsymbol{T}}_{\boldsymbol{1N}} = \hat{\boldsymbol{T}}_{\boldsymbol{4N}}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{2N}} = \hat{\boldsymbol{T}}_{\boldsymbol{3N}}$, depend on the network approximations of the fibers paths $\theta_{1N}(x, y)$ and $\theta_{2N}(x, y)$, respectively. The stiffness matrices $\hat{\boldsymbol{D}}_{\boldsymbol{L1N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L2N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L3N}} = \hat{\boldsymbol{D}}_{\boldsymbol{L4N}}$, instead, depend on the network approximations of the fibers volumetric fraction $V_{fN}(x, y)$, as in Eq. (6.44).

The approximation of the fibers volumetric fraction $V_{fN}(x, y)$ is expressed by means of a 2D KLE built considering the stochastic properties used to perturb $V_f(x, y)$. It follows that $V_{fN}(x, y)$ is:

$$V_{fN}(x, y) = V_f + \sum_{i=1}^{4}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{V_f ij} \tag{6.49}$$

Evenly, $\theta_{1N}(x, y)$ and θ_{2N} are expressed by means of 2D KLEs built considering the same stochastic properties used to perturb $\theta_1(x, y)$ and $\theta_2(x, y)$). It follows that $\theta_{1N}(x, y)$ and θ_{2N} are:

$$\begin{aligned} \theta_{1N}(x, y) &= \theta_1(x, y) + \sum_{i=1}^{4}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_1 ij} \\ \theta_{2N}(x, y) &= \theta_2(x, y) + \sum_{i=1}^{4}\sum_{j=1}^{4}\sqrt{\lambda_i\lambda_j}\phi_i(x)\phi_j(y)\xi_{\theta_2 ij} \end{aligned} \tag{6.50}$$

The boundary condition residual, which is evaluated at a set of N_{CB} collocation points, is:

$$B(\boldsymbol{X_{CBi}}, \boldsymbol{\theta}) = w(\boldsymbol{X_{CBi}}) = 0 \tag{6.51}$$

The network architecture is the one depicted in Figure 6.14. So, 2 neurons for the input layer, 1 for the output layer and 20000 for the hidden layer.
The training process has to learn 20048 internal parameters. In particular, N_S=2500 labeled points, N_{CD}=400 collocation points inside the domain and N_{CB}=400 collocation points along the boundaries are used.
The initial guesses for the output weights are values between -1 and 1. The variables in Eq. (6.49) and Eq. (6.50) are set to 0 such that the initial guess for $V_{fN}(x, y)$ is V_f. Moreover, the initial guess for $\theta_{1N}(x, y)$ is $\theta_1(x, y)$ and for $\theta_{2N}(x, y)$ is $\theta_2(x, y)$.
The maximum number of iterations is fixed to 30, while the tolerance to 10^{-5}.

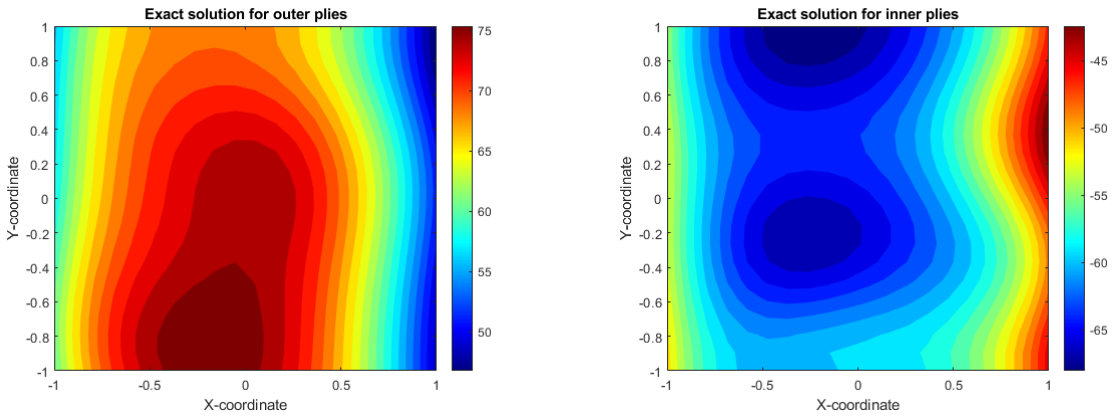
6.7.2. Results

The following results are referred to a laminate made of four plies of thickness $t_p = 0.2$ mm, edge length $l = 50$ mm and loaded by a distributed force $P = 0.5$ MPa. The elastic properties of the composite material are reported in Table 6.3.

The nominal fibers paths, $\theta_1(x, y)$ and $\theta_2(x, y)$, follow a quadratic law. In particular, $[\theta_{1A}, \theta_{1B}, \theta_{1C}]$=[50°,70°,50°] and $[\theta_{2A}, \theta_{2B}, \theta_{2C}]$=[-50°,-70°,-50°]. They are perturbed considering the following stochastic properties:

- correlation lengths $c_X = l$ and $c_Y = l$;
- Exponential correlation function $C(x_1, y_1, x_2, y_2) = e^{\frac{-|x_1-x_2|}{c_X}} e^{\frac{-|y_1-y_2|}{c_Y}}$;
- Variances σ_X^2=5° and σ_Y^2=5°;

It follows that the perturbed fibers paths assigned to the numerical model are the ones depicted in Figure 6.31.



(a) Perturbed fibers distribution for outer plies (b) Perturbed fibers distributions for inner plies

Figure 6.31: Perturbed fibers distributions

Moreover, the volumetric fraction has a nominal value of 0.8 and it is perturbed considering the following stochastic properties:

- correlation lengths $c_X = l$ and $c_Y = l$;
- Exponential correlation function $C(x_1, y_1, x_2, y_2) = e^{\frac{-|x_1-x_2|}{c_X}} e^{\frac{-|y_1-y_2|}{c_Y}}$;
- Variances σ_X^2=0.12 and σ_Y^2=0.12;

It follows that the perturbed fibers volumetric fraction $V_{FP}(x, y)$ assigned to the numerical model is the ones depicted in Figure 6.32.

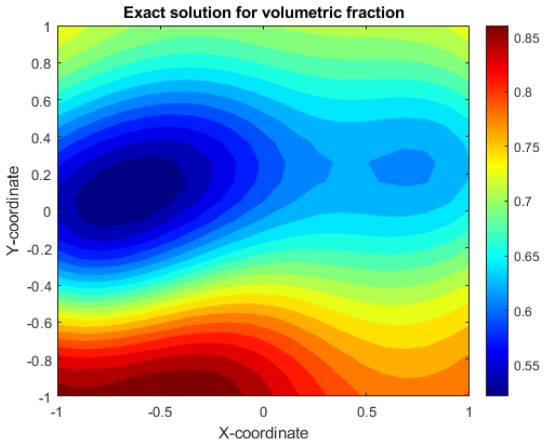
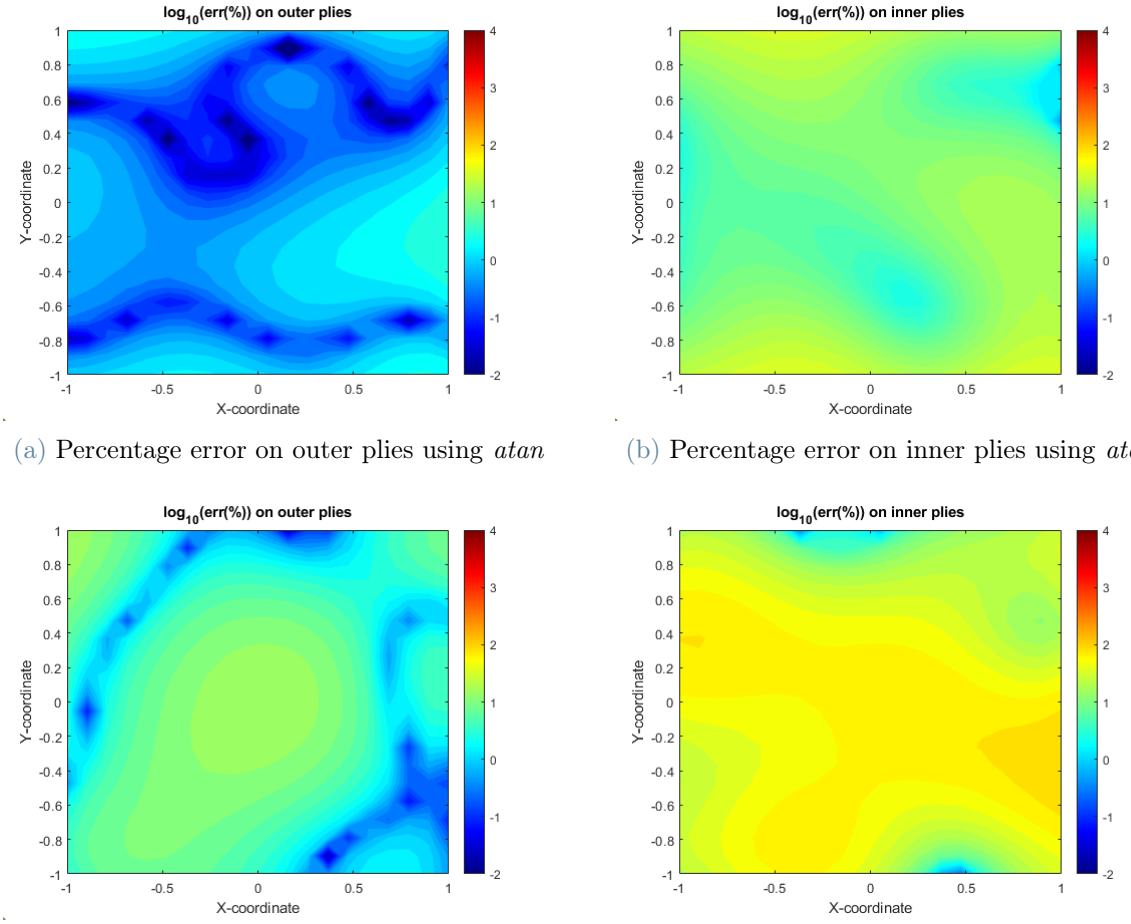


Figure 6.32: Perturbed volumetric fraction

For the same reasons made explicit in the previous cases, the adoption of *sigmoid* as hidden layer activation function leads to very bad results. It follows that only the results

obtained when using *atan* and *tanh* are reported.
In particular, the final percentage error distributions on the fibers misalignment, $err_{\theta_1}\% = \frac{|\theta_{1N}(x,y)-\theta_{1P}(x,y)|}{\theta_{1P}(x,y)}$ and $err_{\theta_2}\% = \frac{|\theta_{2N}(x,y)-\theta_{2P}(x,y)|}{\theta_{2P}(x,y)}$, in logarithmic scale, are depicted in Figure 6.33.



(a) Percentage error on outer plies using *atan*

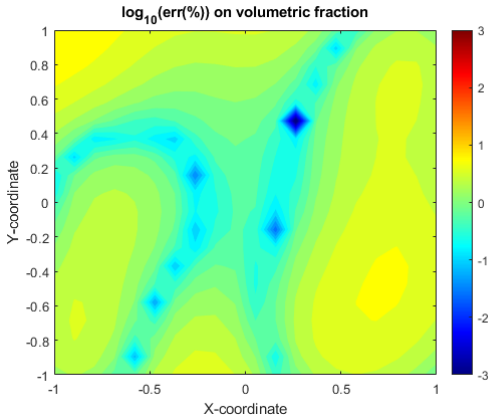
(b) Percentage error on inner plies using *atan*

(c) Percentage error on outer plies using *tanh*

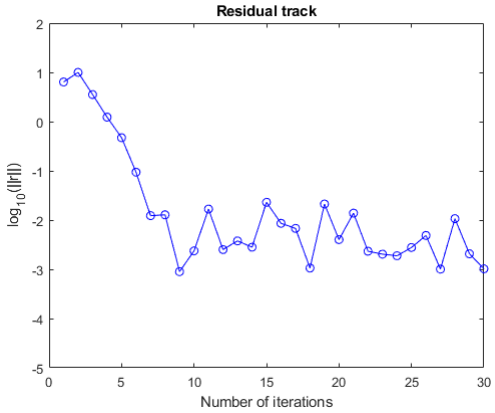
(d) Percentage error on inner plies using *tanh*

Figure 6.33: Final percentage error distributions on fibers misalignment

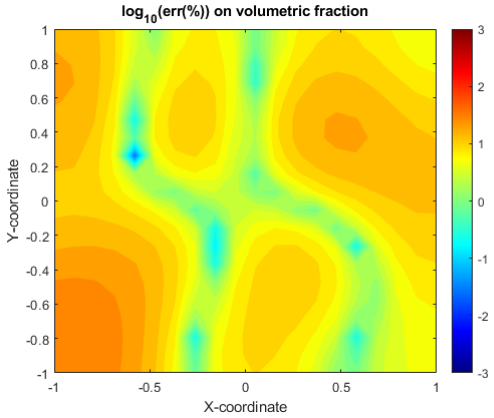
In Figure 6.34, instead, it is reported the final percentage error distribution on the volumetric fraction $err_{V_f}\% = \frac{|V_{fN}(x,y)-V_{fP}(x,y)|}{V_{fP}(x,y)}$, in logarithimic scale, and the evolution of the residual vector norm.



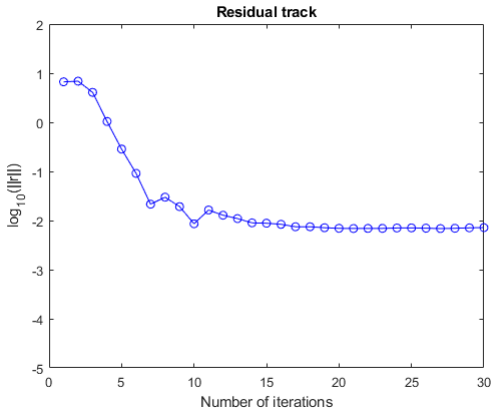
(a) Percentage error on volumetric fraction using textitatan



(b) Residual vector norm evolution using *atan*



(c) Percentage error on volumetric fraction using *tanh*



(d) Residual vector norm evolution using *tanh*

Figure 6.34: Final percentage error distribution on volumetric fraction and residual vector norm evolution

As seen, even if the residual vector norm reaches low values (in the order of 10^{-3}), the percentage errors are much larger than the expected ones. This is similar to issue encountered in Section 6.2. As a matter of fact, in that case the network wasn't able to find the right values of section area and Young modulus, even if the the convergence was reached. Evenly, in this case the network finds a combination of volumetric fraction distribution and fibers paths distributions physically reasonable, since they minimize the residual norm vector, but different from the expected ones. This problem can be addressed to the convergence to a local minimum in the space of the possible solutions, which is a typical issue regarding the gradient-based training algorithms.

As it was done in Section 6.2, a possible solution to this issue could be the addition of

other information regarding the volumetric fraction and/or the fibers distributions in the cost function. Otherwise, another type of optimization algorithm can be used: the genetic algorithm. As a matter of fact, the performance of this type of algorithm is not affected by the presence of local minima in the space of solutions. For this reason, it can be used to find an adequate initial guess for the internal parameters vector, quite close to the global minimum.

7 | Conclusions and future developments

This thesis presented a new method for the detection and quantification of different types of defects affecting Variable-Angle Tow (VAT) structures. The proposed method combines the ability of the Physics-Informed Neural Networks (PINNs) in solving structural inverse problems and the possibility to express the defects distributions by means of the Karhunen-Loeve expansions, which is a tool commonly used for the sensitivity analyses of structures.

Different types of problems are faced to prove the effectiveness of the proposed method. Starting from pristine structures, some of their structural properties were perturbed by introducing defects at known locations. Then, the method was employed to see if it was possible to reconstruct their distributions.

First of all, one dimensional problems regarding the reconstruction of Young modulus and section area distributions for a metallic rod axially loaded were tackled. The results showed that the proposed method works very well the detection of a single distribution is of concern. On the contrary, extra data is needed, besides the set of axial displacements at a certain number of points, when both distributions are unknowns.

Then, two dimensional problems regarding the detection of fibers misalignment, gaps and overlaps affecting symmetric VAT laminates were faced. The results showed that the method works appropriately when only one type of defects distribution is investigated, while the detection of more then one type of defect, at the same time, can be more challenging to be pursued.

On the whole, the results show a good capacity in solving a wide range of problems regarding different structures, metallic or composite, and involving different types of defects. Moreover, since it provides analytical expressions for the defects distributions, it could be a support tool for the realization of numerical models of the defected structures, in order to perform their mechanical characterization. However, its potential seems to be limited by the fact that the method is in difficulty when it is used for the identification of more than one defect distribution at the same time.

Future investigations should be directed towards finding a way to overcome the latter issue. A suggestion is to exploit the genetic algorithms for the generation of an adequate initial guess to overtake the issue of convergence to local minima, which affects the gradient-based training algorithms. Moreover, the extension to more complex structures, like stiffened aeronautical panel, could be of interest.

Bibliography

[1] K. Banerjee. Generalized inverse of matrices and its applications, 1973.

[2] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18: 1–43, 2018.

[3] A. Beakou, M. Cano, J.-B. Le Cam, and V. Verney. Modelling slit tape buckling during automated prepreg manufacturing: A local approach. *Composite Structures*, 93(10):2628–2635, 2011.

[4] A. W. Blom, C. S. Lopes, P. J. Kromwijk, Z. Gurdal, and P. P. Camanho. A theoretical model to study the influence of tow-drop areas on the stiffness and strength of variable-stiffness laminates. *Journal of Composite Materials*, 43(5):403–425, 2009.

[5] M. Buscema. Back propagation neural networks. *Substance Use & Misuse*, 33(2): 233–270, 1998.

[6] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[7] T. Crestaux, O. Le Maıtre, and J.-M. Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, 2009.

[8] A. Darabi and X. Maldague. Neural network based defect detection and depth estimation in tnde. *Ndt & E International*, 35(3):165–175, 2002.

[9] M. W. Davis. Production of conditional simulations via the lu triangular decomposition of the covariance matrix. *Mathematical Geology*, 19(2):91–98, 1987.

[10] P. Duchene, S. Chaki, A. Ayadi, and P. Krawczak. A review of non-destructive techniques used for mechanical damage assessment in polymer composites. *Journal of Materials Science*, 53(11):7915–7938, 2018.

[11] V. Dwivedi and B. Srinivasan. Physics-informed Extreme Learning Machine

(PiELM)–a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020.

[12] I. El Naqa and M. J. Murphy. What is machine learning? In *Machine Learning in Radiation Oncology*, pages 3–11. Springer, 2015.

[13] K. Fayazbakhsh, M. A. Nik, D. Pasini, and L. Lessard. The effect of gaps and overlaps on the in-plane stiffness and buckling load of variable stiffness laminates made by automated fiber placement. In *Proceedings of 15th European Conference on Composite Materials, Venice, Italy*, 2012.

[14] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[15] W. E. Guin, J. R. Jackson, and C. M. Bosley. Effects of tow-to-tow gaps in composite laminates fabricated via automated fiber placement. *Composites Part A: Applied Science and Manufacturing*, 115:66–75, 2018.

[16] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379:113741, 2021.

[17] M. Hassan, A. Othman, and S. Kamaruddin. A review on the manufacturing defects of complex-shaped laminate in aircraft composite structures. *The International Journal of Advanced Manufacturing Technology*, 91(9):4081–4094, 2017.

[18] G.-B. Huang. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274–281, 2003.

[19] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[20] S. Huang, S. Mahadevan, and R. Rebba. Collocation-based stochastic finite element analysis for random field problems. *Probabilistic Engineering Mechanics*, 22(2):194–205, 2007.

[21] A. Jahan, K. L. Edwards, and M. Bahraminasab. *Multi-criteria Decision Analysis for Supporting the Selection of Engineering Materials in Product Design*. Butterworth-Heinemann, 2016.

[22] B. Kim, K. Hazra, P. Weaver, and K. Potter. Limitations of fibre placement techniques for variable angle tow composites and their process-induced defects. In *Pro-*

ceedings of the 18th International Conference on Composite Materials (ICMM18), Jeju, Korea, pages 21–26, 2011.

[23] B. C. Kim, K. Potter, and P. M. Weaver. Continuous tow shearing for manufacturing variable angle tow composites. *Composites Part A: Applied Science and Manufacturing*, 43(8):1347–1356, 2012.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*, 2014.

[25] X. Li, S. R. Hallett, and M. R. Wisnom. Modelling the effect of gaps and overlaps in automated fibre placement (afp)-manufactured laminates. *Science and Engineering of Composite Materials*, 22(2):115–129, 2015.

[26] R. Lincoln, P. Weaver, A. Pirrera, and R. M. Groh. Optimisation of imperfection-insensitive continuous tow sheared rocket launch structures. In *AIAA Scitech 2021 Forum*, page 0202, 2021.

[27] Y. Lu, L. Ye, Z. Su, L. Zhou, and L. Cheng. Artificial neural network ann-based crack identification in aluminum plates with lamb wave signals. *Journal of Intelligent Material Systems and Structures*, 20(1):39–49, 2009.

[28] H. Luo, H. Zeng, L. Hu, X. Hu, and Z. Zhou. Application of artificial neural network in laser welding defect diagnosis. *Journal of Materials Processing Technology*, 170 (1-2):403–411, 2005.

[29] T. A. Mara and W. E. Becker. Polynomial chaos expansion for sensitivity analysis of model output with dependent inputs. *Reliability Engineering & System Safety*, 214: 107795, 2021.

[30] A. Marouene, R. Boukhili, J. Chen, and A. Yousefpour. Effects of gaps and overlaps on the buckling behavior of an optimally designed variable-stiffness composite laminates–a numerical and experimental study. *Composite Structures*, 140:556–566, 2016.

[31] C. J. McInnes, R. Lincoln, A. Pirrera, B. C. Kim, and R. M. Groh. On the finite element discritization of continuous tow sheared structures. In *AIAA SCITECH 2022 Forum*, page 2598, 2022.

[32] S. S. Nimbal, M. M. Banker, A. Roopa, B. Varughese, and R. Sundaram. Effect of gap induced waviness on compressive strength of laminated composites. *Materials Today: Proceedings*, 4(8):8355–8369, 2017.

[33] A. Pagani and A. Sanchez-Majano. Stochastic stress analysis and failure onset of variable angle tow laminates affected by spatial fibre variations. *Composites Part C: Open Access*, 4:100091, 2021.

[34] Z. Pan, L. Zhang, and K. Liew. The use of curvilinear fibers for enhancement of progressive failure performance of perforated composite panels. *Composite Structures*, 288:115424, 2022.

[35] B. S. G. Pernambuco, C. R. Steffens, J. R. Pereira, A. V. Werhli, R. Z. Azzolin, and E. d. S. D. Estrada. Online sound based arc-welding defect detection using artificial neural networks. In *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, pages 263–268. IEEE, 2019.

[36] K.-K. Phoon, S. Huang, and S. T. Quek. Simulation of second-order processes using karhunen–loeve expansion. *Computers & Structures*, 80(12):1049–1060, 2002.

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

[38] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv Preprint ArXiv:1609.04747*, 2016.

[39] A. R. Sanchez-Majano, A. Pagani, M. Petrolo, and C. Zhang. Buckling sensitivity of tow-steered plates subjected to multiscale defects by high-order finite elements and polynomial chaos expansion. *Materials*, 14(11):2706, 2021.

[40] D. U. Shah. Natural fibre composites: Comprehensive ashby-type materials selection charts. *Materials & Design (1980-2015)*, 62:21–31, 2014.

[41] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[42] B. Sudret and A. Der Kiureghian. *Stochastic finite element methods and reliability: a state-of-the-art report.* Department of Civil and Environmental Engineering, University of California . . . , 2000.

[43] L. Wang. *Karhunen-Loeve expansions and their applications.* London School of Economics and Political Science (United Kingdom), 2008.

[44] P. Wei, C. Tang, and Y. Yang. Structural reliability and reliability sensitivity analysis of extremely rare failure events by combining sampling and surrogate model methods.

Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 233(6):943–957, 2019.

[45] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[46] W. Woigk, S. R. Hallett, M. I. Jones, M. Kuhtz, A. Hornig, and M. Gude. Experimental investigation of the effect of defects in automated fibre placement produced composite laminates. *Composite Structures*, 201:1004–1017, 2018.

[47] Z. Wu, P. M. Weaver, G. Raju, and B. C. Kim. Buckling analysis and optimisation of variable angle tow composite plates. *Thin-walled Structures*, 60:163–172, 2012.

[48] D. Xiu and G. E. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[49] C. Yan, R. Vescovini, and L. Dozio. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. *Computers & Structures*, 265:106761, 2022.

[50] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

[51] W. Zhao and R. K. Kapania. Buckling analysis and optimization of stiffened variable angle tow laminates with a cutout considering manufacturing constraints. *Journal of Composites Science*, 6(3):80, 2022.

List of Figures

List of Tables

Acknowledgements

In conclusion, I would like to mention all the people that helped me to reach this milestone.

I would like to say thanks to my supervisor Riccardo Vescovini, who helped me during the realization and the writing of this thesis, by giving precious advice.
Moreover, the assistance provided by my co-supervisor Cheng Angelo Yan was greatly appreciated.

I am deeply grateful to my family, that unconditionally sustained me in each step of my student career. Especially to my mom, who always listened to my vents and helped me in the moments of discouragement.

I would like to say thanks to my university mates for the beautiful moments experienced during these 5 years.

In the end, I would like to say thanks to myself, hoping this is the starting point for brilliant and successful professional career.