Analysis of composite laminates using Physics Informed Neural Networks and Extreme Learning Machine

POLITECNICO
MILANO 1863

Author: Iván Barcelona Moreno - 912505
Supervisor: Riccardo Vescovini
Co-supervisor: Cheng Angelo Yan

A thesis presented for the *Laurea Magistrale* (MSc) in
Space Engineering

Dipartimento di Scienze e Tecnologie Aerospaziale (DAER)
Politecnico di Milano
Academic Year 2020/2021
Milano, Italy
July 23^{rd}*, 2021*

Contents

List of Figures

List of Tables

Acknowledgements

I would first like to express my sincerest gratitude to Professor Riccardo Vescovini, my supervisor, whose vast knowledge and advice helped me develop and enjoy the realization of this work. My special thanks to Cheng Angelo Yan, whose advice on coding were always welcome and helpful.

Finally, to my parents Juan and Inma, who made me the person I am right now and whose inconditional support made me keep going in this way.

Abstract

The analysis of multilayered composite structures presents an ever increasing challenge with the appearance of new levels of complexity within material science. Proper tools are in constant development to give researchers the capabilities to better understand the nature of materials and how to use them. A novel methodology in the field of Machine Learning (ML), known as Physics Informed Neural Networks (PINN) has been implemented. This architecture is employed to approximate the solution of boundary-value collocation problems formulated in compliance with the Sublaminate Generalized Unified Formulation (S-GUF). The novelty of the approximation resides in the joint application of PINN and the Extreme Learning Machine (ELM) algorithm, which is based on the solution of a minimum-norm least squares linear system to fulfill the physical constraints imposed by the problem. An additional layer of novelty makes it the first attempt to use these techniques in a combined fashion to a variable-kinematics formulation, without previous work reported in the area. The aim of this approach is to gain access to a new tool that allows the analysis of complex multilayered plates with different variable kinematics models in an efficient way, allowing the comparison of methods and models.

A special attention is given to the comparison and stability of strong and weak-form formulations within this framework. A strong approach is developed by extracting a system of partial differential equations from the expression of the Principle of Virtual Works, that are imposed across all points in a collocation grid to enforce equilibrium and compatibility conditions. Certain numerical issues are reported along the work, motivating the development of new ways to overcome them. The coupled influence of network parameters and other factors given by the geometry or the approximation is studied thanks to a detailed sensitivity analysis. Finally, the initial implementation made for rectangular plates is extended to shells, and plates of arbitrary shapes, what makes it a versatile tool that overcomes the limitations of other methodologies only suited to solve simple problems.

Sommario

L'analisi delle strutture composite multistrato rappresenta una sfida sempre più grande con la comparsa di nuovi livelli di complessità nella scienza dei materiali. Strumenti adeguati sono in costante sviluppo per dare ai ricercatori le capacità di comprendere meglio la natura dei materiali e come utilizzarli. È stata implementata una nuova metodologia nel campo del Machine Learning (ML), conosciuta come Physics Informed Neural Networks (PINN). Questa architettura è impiegata per approssimare la soluzione di problemi di collocazione dei valori limite formulati in conformità con la Formulazione Unificata Generalizzata del Sublaminato (S-GUF). La novità dell'approssimazione risiede nell'applicazione congiunta di PINN e dell'algoritmo Extreme Learning Machine (ELM), che si basa sulla soluzione di un sistema lineare dei minimi quadrati a norma minima per soddisfare i vincoli fisici imposti dal problema. Un ulteriore livello di novità lo rende il primo tentativo di utilizzare queste tecniche in modo combinato a una formulazione di cinematica variabile, senza precedenti lavori riportati nell'area. Lo scopo di questo approccio è quello di ottenere l'accesso a un nuovo strumento che permetta l'analisi di piastre complesse multistrato con diversi modelli di cinematica variabile in modo efficiente, permettendo il confronto di metodi e modelli.

Un'attenzione speciale è data al confronto e alla stabilità delle formulazioni forti e deboli in questo quadro. Un approccio forte è sviluppato estraendo un sistema di equazioni differenziali parziali dall'espressione del Principio dei Lavori Virtuali, che sono imposte attraverso tutti i punti in una griglia di collocazione per imporre condizioni di equilibrio e compatibilità. Alcuni problemi numerici sono riportati lungo il lavoro, motivando lo sviluppo di nuovi modi per superarli. L'influenza accoppiata dei parametri di rete e di altri fattori dati dalla geometria o dall'approssimazione è studiata grazie a una dettagliata analisi di sensibilità. Infine, l'implementazione iniziale fatta per piastre rettangolari è estesa a gusci e piastre di forme arbitrarie, il che lo rende uno strumento versatile che supera i limiti di altre metodologie adatte solo a risolvere problemi semplici.

1 Introduction

For the last few decades, composite materials have become an outstanding bet in material development and innovative applications for researchers and industrial agents. The capability to tailor their properties for specific applications has made them to be a remarkable consideration along with other traditional materials for almost any technological application. Lightweightness and directional properties make them largely efficient for mass sensitive applications, herein the interest from the aerospace sector to develop new materials and analysis techniques that allow to evolve the design complexity to a new level. Composites possess a very attractive combination of stiffness, toughness, durability and are lightweight, that in conjunction makes them specifically suitable for many applications. As its own name indicates, they come from the combination of more than one material, in the majority of cases, a fiber that provides most of the mechanical properties, and a matrix, typically a resin-like material (i.e. *epoxy*), that bonds fibers together. However, this combination of components make it critical to understand well how they are joined together to confer the composite with its properties: parameters like fiber size, density or orientation, as well as ply-to-ply stacking sequence when fabrics are used to create thicker elements play a critical role in their design and manufacturing [4].

The development of industry towards a more automatized and high-tech concept has made it possible for the introduction of composites and their manufacturing techniques at an increased rate, making them almost standard in some applications. An interesting example of this integration is the recently finished *Airbus A-380* project, which included some novel composite applications thanks to the experience gained on previous programmes. This model employed a Carbon Fibre Reinforced Plastic (CFRP) composite central wing box which meant over a tone and a half weight saving compared to traditional alluminium alloys. Weight of the components and need to withstand large in-flight stresses also let the introduction of CFRP keel beams, rudder, horizontal stabilisers and elevators [5]. Novel uses of computerized techniques include *Automated Tape Laying* (ATL), where unidirectional tapes are directly laid onto a mould with a roller system [6]. Floor beams and rear pressure bulkheads were manufactured following *Resin Film Infusion* (RFI) and *Automated Fibre Placement* (AFP), completting such an idilic case of composites application.

Furthermore, composites have revolutionized the space sector due to their tailorable properties and capacity of withstanding space environment. In here we find a predominance on the concept of *Fiber Metal Laminates* (FML), consisting on the reinforcement of aluminun sheets with layers of fiber adhesives. Corrosion resistance, damage tolerance or thermal insulation are some of the inherent disadvantages to metals that this combination manages to overcome. Glass and kevlar fibers have seen a growing application in satellite antenna technology due to favorable transmission loss and electrical conductivity. Additionally, aluminum core honeycomb with reinforced carbon epoxy laminates has proven effective to payload bay doors due to high specific strength, damage tolerance, fatigue and crack resistance, critical properties in direct exposure to space high-velocity fragment collisions. Furthermore, composite materials have found a strong niche in launch vehicles, where there has been a great research on *Advanced Grid Stiffened Structures* (AGS) [7] in the last few years. Carbon epoxy filament winding techniques to manufacture payload shrouds resulted in more than 60% weight reduction and 80% fabrication time savings according to some authors [8].

A more terrestrial application that has gained enormous support in industry is that of the automotive sector. With combustion engines pushed near their limit efficiency, as well as the appearance of electric cars, composite material lightweightness constitutes a property that can

lead to potential fuel consumption and manufacturing costs reduction, making it actually very attractive for companies. Traditional actors of the sector are using composites in high-class cars to boost performance and accomplish emission regulations concurrently (as well as adding certain "luxury" to the brand). In comparison to steel usage, they add an advantage in the drawing angles of metal stamping, resistance to corrosion and scratches, and an intrinsic safety due to larger energy absorption against impacts. While CFRP and GFRP (*Glass Fiber Reinforced Polymer*) are the most common for high loading applications, other composites are employed for not-so demanding specifications: GMT (*Glass Mat Reinforced Plastics*) and LFT (*Long-fiber Reinforced Thermoplastics*) are increasingly used in spare tire holders or front/back ends [9].

All of the previous make it clear that material science is on an exponential development and will most probably continue like that in the future. The use of composites or their combination with metals and others materials in a multilayered fashion seems to be a recurring trend in research for the past few years, aiming to develop materials with better properties for same and new applications. To make it possible, the analysis and understandment of their behaviour must also share that evolutionary path, and the look for better tools and formulations to accomplish this task is a priority.

1.1 State of the art in analysis tools

There has been a great interest on the development of computational tools for the analysis of multilayered panels, noted in the large amount of publications in the past decades [10] - [11]. *Kinematic theories* are introduced as a modelization of the behaviour of field variables, in order to undergo a structural analysis. Two major families were introduced by Reddy [12] in the context of composite multilayered structures referred as *Equivalent Single Layer* (ESL) and *Layer-Wise* (LW) models. ESL gives a simplified description of the laminate, introducing an anisotropy that describes as a single entity the stackup of different plies. LW, on the contrary, treats each layer independently, allowing for a piece-wise description of the kinematic quantities that resembles better its real behaviour. An enhancement to ESL descriptions is given by Carrera [13] in the form of a *zig-zag* theory, an additional term that includes an alternating sign through-thickness function in order to introduce some heterogeneity inherent of composites in the transverse behaviour. Regarding purely ESL descriptions, two well-known examples are the *Classical Laminate Theory* (CLT) and the *First-Shear Deformation Theory* (FSDT). The first is also referred as *Kirchhoff Plate Theory*, although this definition usually accompanies non-multilayered panels. It introduces a series of hypothesis that hold for its conception of the plate transverse behaviour [14]:

- Transverse normals to the midsurface [1] remain straigth after deformation, not allowing for transverse shear deformation effects to be present.
- Transverse normals do not elongate.
- Rotation occurs such that transverse normals remain perpendicular to the midsurface.

On the other hand, FSDT is analogously referred as *Reissner-Mindlin Plate Theory*, and removes the first Kirchhoff assumption regarding the introduction of deformation shear [15]. Kirchhoff and Mindlin rely on the plane stress assumption, that makes use of a reduced constitutive law [16]. Both these theories are specially important as most simplified versions of more complex descriptions introduced by so-called *higher order theories*. Some of them are

[1] Understood as the surface at the midspan of the thickness along the plate transverse direction.

used as standalone models for some very specific phenomena, *Reddy's third order theory* is a good example, showing a good applicability to analyze delamination problems as stated by Szekrényes [17]. These theories are differentiated from the order of the polynomial expansion they use.

All of the previous are postulated as *axiomatic approaches*, a simplification of the description that assumes certain hypotheses to represent the behaviour in the thickness direction, while focusing on modelling the in-plane characteristics of the panel. In the same way that LW theories describe an heterogeneous setup of composite layups (i.e. fiber orientation, material, layer thickness...), the interest to develop analytical tools for sandwich panels [18] motivated the definition of *sublaminate*, performed by Reddy [19], as an entity grouping several plies sharing common properties or a functionality (*face*/*core*). A remarkable attempt to introduce a unified framework was made by Carrera *et al* [20] with *Carrera's Unified Formulation* (CUF), where he introduced an indicial notation that could be used to describe the governing equations of very different models in a common framework. The major inconvenient is that the polynomial order selected for the through-thickness description has to be equal for all unknowns, what reduces the freedom of choice for the analysis. This constraint motivated the appearance of the *Generalized Unified Formulation* (GUF) introduced by Demasi [21], where different orders of approximation may be taken for different field variables. Further work by Demasi allowed to describe different displacement components by either LW/ESL theories [22] in what are known as *partially Layer Wise* models. However, an inherent limitation of both CUF and GUF still remained present, once a model theory and expansion order were selected, that same description had to be applied to every ply in the domain. This motivated the recent appearance of a third framework, known as the *Sublaminate Generalized Unified Formulation* (SGUF) by D'Ottavio [16], that recovered Reddy's sublaminate definition [19]. In it, each field variable can be described independently with GUF, and different descriptions can be given to each sublaminate, allowing for an unprecendented tailorability. All these theories are usually referred as *variable-kinematic* theories, as they allow to select the proper expansion order for each field variable.

With the kinematic description of the panel, different formulations can be used to derive the governing equations. To this respect, there is a differentiation between *displacement-based* and *force-based* formulations. In a displacement framework, displacement components are the variables of interest to directly recover from the solution of the linear problem. The force-based approach is also referred as stress-based, the dominant terms of the formulation are the stresses, that become unknowns to the problem. Using the stress field allows to obtain a better description in general terms of the stress components in the transverse direction. Both constitute the basis for the development of new formulations, and specially displacement-based have found a niche of application in the field of Finite Element Analysis (FEA). Some authors like Carrera do specifically apply FEA to the field of composite materials [23], but not only restricted to purely displacement-based and force-based formulations. A third branch comes up from the combination of previous contributions, in the form of a *mixed formulation*. This is not a unique term, but encompasses different modelling possibilities were displacements, rotations or stress-related quantities are considered. Kiendl *et al* [24] introduce a formulation of this kind to alleviate shear-locking effects on thin isotropic Reissner-Mindlin plates. This effect makes the plate to be unable to bend due to a spurious numerical constraint introduced in the first derivative of the rotation. This work was later extended in the context of composites by Pavan and Rao [25]. By explicitely introducing shear forces, they were able to solve locking problems with an equivalent performance to specifically developed formulations with this purpose, what makes this implementation an interesting option in similar contexts.

The authors for both CUF [20] and GUF [21] introduce the displacement fields in the context of a variational statement that makes use of virtual quantities. A pure displacement definition is given by the *Principle of Virtual Works* (PVW), while specially Demasi focuses a substantial amount of his work on implementing the *Reissner's Mixed Variational Theorem* (RMVT). The latter is an alternative that allows to introduce independent approximations in the transverse direction, and in particular to the out-of-plane stresses, assigning different orders of expansion [21], what can be a useful tool in fulfilling interlayer equilibrium. Two approaches can be followed for its solution, distinguished by the form those governing equations take. The *strong-form* formulation solves the differential form of the equations, thus requiring of a numerical technique able to approximate derivatives [26]. On the other hand, the *weak-form* formulation introduces an approximation to reduce the differentiability order of the equations [27], and in particular the Finite Element Method (FEM) is based on this weak form, extensively reported by Zienkiewicz in its book [28].

Different solution strategies are available in literature. The development of ever increasing complex computational tools has led to an extensive utilization of *Finite Element* models to study problems of any kind. Composite multilayered plates pose an inherent complication for analysis in terms of their in-plane and out-of-plane strains coupling and presence of anisotropy, leading to very few closed solutions. The application of FEA tries to overcome this limitation in an approximated way with the generation of 2D/3D elements that discretize the domain, and where the governing equations can be imposed at a local level. Carrera extensively reports the possibilities of application of finite elements and the mathematical foundation of the models used for mechanical analysis in the context of CUF [23]. On the other hand, Ochoa [29] develops finite element models both for CLT and FSDT, with an emphasis on the assembly of elements that leads to continuity of nodal displacements and the balance of nodal forces. Purely stress-based formulations are also reported, as it is the case of Han and Hoa [30] who use the transverse stresses and a partial stress field parameter to enforce continuity in the laminate thickness direction. The great advantage of FEA over other methods is that it allows to study highly complex domains. However, it is often slow provided that computational time depends directly on mesh size and oftenly grid independence studies have to be performed to assess that the mesh definition is indeed causing no effects on the fidelity of the solution. For this reason, it usually requires experienced handling and extensive computational resources, what makes it uninteresting for a preliminary study of a problem.

Another family of analysis tools featured in literature with an inherent power for structural analysis is that of *collocation methods*. They allow for the numerical solution of differential equations by choosing a set of candidate points inside the domain where those equations are specifically imposed, and then generalizing the solution to the rest of the domain. It is indeed a discrete method and aims at solving the problem both in the spatial and temporal domain (if required), from a set of initial conditions. Collocation methods can be extended to any kind of situations where a set of differential equations solution has to be approximated, as it is the case of strong-form formulated structural problems. Pavan and Rao mention in [25] the potential of this family of methods to implement his shear mixed-formulation, and in particular the relatively new Isogeometric Analysis (IGA), first stated by Hugues [31]. This particular collocation method has been studied by several authors [32] [33] [34] , able to represent an exact geometry model from a set of basis functions generated by Non-Uniform Rational B-Splines (NURBS). The unknown field variables are expressed as linear combination of NURBS, which make indeed a complete and admissible set in a similar fashion to shape functions in FEA. The discretization of those variables across the domain is then introduced into the governing equations, leading to

a system that can be solved numerically. In [25], the author uses a particular category of IGA as solution method, the *Galerkin-IGA* method. This is a suitable selection, provided Galerkin method (also referred as Bubnov-Galerkin) is meant to approximate the unknown functions in the governing equations by using a set of global functions. Because of the approximation, both Galerkin and IGA-Galerkin require the definition of residuals over the domain, that are *collocated*, set to zero in order to constraint and close the numerical problem. Galerkin method has been extended by Gulizzi *et al* [35] to the domain of multilayered composite plates, and in particular to be used for layer-wise theories, considering the piece-wise transverse behaviour of the solution fields. Additionally, Chen *et al* [36] used a precursor to IGA analysis called Mesh Free Method (MFM) along with Galerkin methodology to perform a free vibrations analysis of composite plates. A set of discretized points both at the domain interior and the boundaries were used to approximate a set of displacement shape functions used then by Galerkin to solve the governing equations. An additional contribution to the field of collocation methods is reported by Ferreira *et al* [37], where he applies an interpolating wavelet basis to solve for a composite plate under bending described by FSDT theory. The major aim of this work was to exploit this family of approximating functions, which are mentioned effective at displaying local behaviours, an interesting contribution to take into account introduced singularities such as concentrated loads, or the presence of inner holes or apertures.

The previous methodologies are specially relevant when working in the context of strong-form formulations, having a system of governing equations to deal with. However, a counterpart to Galerkin in a weak formulation environment is found within the *Ritz method*, that belongs to a class of approximating techniques based on the variational formulation. This method can be used either in the context of displacement-based or force-based formulations. Early works on this topic includes that of Young [38] that obtained the natural frequencies for the vibration of metallic rectangular plates, or Martin and Leissa [39] that applied the method a few decades later to composite sheets in order to test the mechanical effects of varying the fiber spacing. A more recent contribution is given by Vescovini *et al* [40], that particularizes the analysis of highly anisotropic plates in the context of CLT theory and Demasi's GUF, and by D'Ottavio *et al* [41] that introduces an interesting implementation of the Ritz method in the context of SGUF formulation for the analysis of piezoelectric composite plates. In both cases, the underlying idea of the Ritz method is to turn the differential problem into an algebraic one by imposing the stationarity condition of a functional, typically the total potential energy. A major limitation of Ritz is on its difficult applicability to domains with complex shapes, while it can perform exceptionally for simple shapes with few degrees of freedom.

Another alternative, in the same line of collocation methods has been studied and was originally proposed by Bellman *et al* [42], called the *Differential Quadrature Method* (DQM). This method can be applied to boundary-value problems, particularly to some applications where numerical solutions for PDEs are required only at a few points. In such cases, its advantage over FEA is purely computational, provided that in the latter a large discretization of the domain would be required to obtain comparable accuracy when only those interest points would be kept. Differential quadrature allows to obtain derivatives with respect to physical coordinates at particular points in the domain expressing such derivatives as a weighted sum of functional values. A later modification of this method was proposed by Shu [43], known as Generalized Differential Quadrature (GDQ) that generalizes the determination of the weighting coefficients (problem unknowns) to the analysis of a high order polynomial approximation. It showed however an inherent limitation in solving for fully clamped cases, and the impossibility to represent free edges. These issues have been corrected in later works [44], that extended its application to general boundary conditions. Tornabene *et al* [45] apply the method to both strong and

weak-form formulations for the free vibrations analysis of laminated plates and shells, while Fantuzzi and Tornabene [2] use DQ methods in combination with IGA, used in order to map non-linearities in complex shapes. A later implementation to use GDQ locally was extended for doubly-curved laminated shells by Tornabene [46].

A field of novelty has opened recently in computer science with the emergence of Machine Learning (ML) techniques [47]. Their applications to scientific purposes have been present for a few years [48] but it has been during the last decade when they have undergone a rapid growth in complexity. It is of particular interest the evolution of a submodel of ML, called *Artificial Neural Networks* (ANN), computational networks biologically inspired by the connections and data transfer of organic being neurons. They allow to identify and exploit hidden patterns in training data, that are then used to analyze unknown data never seen before. It is done through a series of connections that transfer input data to the network units called *nodes*, and organized in *layers* where a mapping is applied thanks to an activation function. A progressively more complex information is grasped by the network after each layer, until they reach an output layer where the answer to the prediction is given. Their great potential is in their high speed processing, that has made them quite competitive in regression and statistical problems regarding traditional counterparts. They possess very good properties in terms of adaptivity, self-learning, non-linearity and input/output mapping, which make them able to generate models of complex natural systems starting from large inputs [49].

In the context of structural analysis, it is mentioned by Rogers [50] the interest on developing neural networks to solve structural optimization problems requiring several costly iterations. The time and resources allocated to training the network made it worth their use for complex problems as opposed to classical FEA techniques. Their major advantage as reported by Lagaros [51] is that once the training is completed, only a trivial computational effort is required to get an approximate solution. This author also reported a learning improvement methodology used over feed-forward networks, a specific architecture suffering from ill-conditioning problems. The evolution of computational speed and power lead to the generalization of *deep learning* techniques [52] [53] [54], a branch of ML used in ANN based on a set of algorithms aimed to simulate complex problems with a high abstraction level. Among recent developments of special interest, there are those first publications by Raissi *et al* [55] [56] focused on the use of neural networks for the solution of linear partial differential equations. Further publications [57] [58] extended the framework to solve non-linear differential equations with special attention in the field of fluid mechanics. The latter contribution [58] introduced the concept of *Physics-Informed Neural Networks* (PINN), a category of ANN suited for supervised learning applications and capable of fulfilling any given physical law thanks to the introduction of additional terms in the loss function representing that fulfillment of the governing equations and boundary conditions. The author reported the use of deep neural networks (several intermediate layers) in a different fashion to traditional ANN, where they are not used as universal function approximators but make only one of the contributions in the minimization of the loss functions while the rest is given by the implied physics. Automatic differentiation is reported as a key tool to differentiate neural networks with respect to the inputs and be able to manage the differential nature of governing equations. This methodology was reported by Baydin *et al* [3] as a differentiation technique used as counterpart to symbolic and numerical differentiation, and with no errors introduced in the process, thus making the differentiation to be exact thanks to a set of intermediate operations. The combination of these tools in ANN set the basis for an ongoing development of new techniques, both in structural analysis and other fields. A particular application of PINN to linear elasticity was made by Hagighat *et al* [59], with the solution of the displacement field in a Kirchhoff plate under bending. The author

outlines a better performance is obtained when a mixed-formulation is used as opposed to a pure displacement-based approach, and introduces a relevant element of novelty by splitting a densely connected network into separated uncorrelated networks ruled by independent parameters. An additional work is reported by Li *et al* [60] on the application of PINN in the context of an energy method based on the minimization of the total potential energy (used as the loss function) and the numerical integration of the energy across the domain at discrete locations. It is influenced by a previous work by Samaniego *et al* [61], where an energy approach is also implemented but in the context of a purely data-driven ANN. In both cases, a relevant element is found on the enforcement of essential boundary conditions by the introduction of a penalty contribution to the potential functional.

The evolution of neural networks opened the possibility not only for new architectures or solution methods based on the underlying physics, but to the appearance of new training methodologies. The most well-known training algorithm is the *gradient descent*, used in [59], that modifies the network connections to progress in the direction of minimization of the loss function for each batch of training. Other training algorithms exist, and are majorily focused on using the gradient or any other derived quantities as metrics of the progression of training. Huang *et al* [62] reported a new training methodology called *Extreme Learning Machine* (ELM) that substantially modified the training process to that of solving a linear system by a Least Squares method. The proposed methodology consisted on randomly initializing the values of network weights and biases using a Single Layer Feedforward Network (SLFN) architecture (with only one intermediate/hidden layer), and then obtaining the values of the output weights by fitting the system. The motivation to use an SLFN was also introduced by Huang in a later publication [63] where he explicited that a network of these characteristics was able to learn an amount of distinct observations equal to the number of hidden nodes. This outbreak has motivated the exploration of this method by other authors, provided its inherent computational simplicity, as it only requires the solution of the linear system and the architecture is constrained to a single hidden layer. Xiao [64] extends the same formulation developed by Huang to a two-layer feedforward network called TELM, and paves the way for its generalization to arbitrarily deep networks. Due to its novelty, there are no reported specific applications on structural analysis that use the ELM training technique.

The present contribution focuses on extending the applicability of SGUF theory in the context of Machine Learning (ML), and in particular by the application of a combined PINN-ELM framework for the first time. A collocation method is used to discretize the associated boundary-value problem to composite multilayered panels, then approximated by the neural network both in the context of *linear static* and *free vibrations* analyses. Two techniques are developed and compared, the first based on the derivation of strong-form governing equations from the Principle of Virtual Works (PVW) which uses the collocation grid to locally evaluate the equations, and a (weak) energy method focused on minimizing the definition of total potential energy and embedded into the network through a numerical integration technique and the use of a penalty factor methodology. This approach allows to extend the analysis from simple shapes to general geometries, more easily than other implementations such as Ritz and Galerkin methods, but without the inherent complexity of FEA. It is also meant to set foot on the generalized introduction of ML techniques in the context of linear analysis, and to be a necessary first step for the future evolution into more complex methods.

1.2 Work organization

The work is organized as follows. Present Section [1] introduces an state-of-the-art review on structural analysis and solution techniques. Section [2] gives a fundamental insight on basic ML concepts, as well as major neural network categories, training algorithms and architectures. Section [3] is meant to introduce SGUF theory in a detailed way, along with the variational principle used to set up the governing equations, and the detailed formulation of the two methods implemented. Section [4] refers to PINN and ELM concepts in detail, and to cover the topic of network hyperparameters influence and training process. Section [5] focuses on presenting major results and comments derived from the application of the method to both invented and literature cases, along with some sensitivity analysis on relevant parameters. Section [6] closes the dissertation with a brief summary on the developed topic, and Section [7] outlooks towards future work and possible implementations.

2 Machine Learning: an overview

The concept of *Machine Learning* has sounded for the last few years in an increasing range of applications as an opportunity to broaden knowledge in complex problems where the amount of available data is large. Its most extended branch for scientific applications has been mentioned to be that of *Artificial Neural Networks*, entities aimed at recognizing complex patterns in the data and make predictions based on this understanding. In order to be trained, a *loss function* is minimized, that includes the difference with the predicted values in the form of input/output data pairs. The cost function minimization trains the network, but to do so, an exact or approximated solution (from experimentation or other numerical models) must be known for training, this is what we usually refer to as *supervised learning*. On the other hand, there are applications where it is most beneficial to use an *unsupervised learning* scheme, typically for clutering and categorization when unlabeled data is available, but it has very few scientifical applications and falls better within data analytics kind. Furthermore, there is an special focus given currently to what is known as *Reinforcement learning* (RL), an innovative approach in many control algorithms, applied to improve the knowledge of the environment or to optimize action policies, and where the network is capable of learning progressively more complex features to better approximate its predictions. An schematic of these categories is given in Fig.(1):

Figure 1: Learning algorithm categories used in ANN.

ANNs can be classified into several categories. Major distinctions are between:

- **Feed-forward neural network (FFNN)**: information transmission occurs only in one direction, from input nodes to hidden layers and then to output nodes. It acts as a classifier, in the way that no feedback is reported back between layers. The depth of the network (number of hidden layers) depends on the application, and the connection between layers can be distinguished into *dense* and *sparse* whenever there is a connection between each node of the present layer and all nodes in the previous or not. Furthermore, specific architectures inside FFNN such as SLFN had great success in mapping low complexity features from input to output while keeping an extraordinary computational speed.

- **Backpropagation networks (BPN)**: make use of the same feed-forward architecture network. In this case, the input data is feeded forward into the network up to the output layer. Then, the expected outputs from the available dataset are used to compute an error δ, that is propagated back into the network to previous layers. Regarding the loss function, this method helps to calculate its gradient with respect to all network weights.

- **Deep Neural Networks (DNN)**: it is placed inside the category of Deep Learning (DL), a methodology that enables more complex ways of connecting layers and allows automatic extraction of features. It is one of the fields that most interest awakes in the scientific community both due to its good fitting for *Big Data* applications (speech recognition, automatic recommendations...) as well as being a crucial part in AI development. Convolutional networks are one of the most common examples, and are mainly used for image recognition. They have multiple layers of different types, and extend in

space by sharing weights, such that it is capable of recognizing features more and more complex from an image after each convolutional layer. Their application to robotics is very interesting as an assistant part in sensor image recognition to feed a more complex Reinforcement Learning (RL) that takes decisions based on sensor information.

- **Physics Informed Neural Network (PINN)**: in this type of feed-forward network, the loss function to be minimized has a more complex form, that includes a physical related knowledge of the problem. Considering a system that has been modelled based on some governing equations, with its own boundary and initial conditions that can be derived from the setup, PINN architecture considers the direction of minimization of the loss function such that this physical rules are rewarded. It is susceptible to any training algorithm of preference and combination of network parameters, while it is possible to apply it also in deep learning.

Out of the architecture, what really makes the difference between neural networks is the learning algorithm they employ to fulfill their purpose. The learning is ruled by the minimization of a *loss function*, that measures their performance for a given dataset. Such fuction is composed by an error metric, that measures the overall discrepancy with respect to an expected behaviour, and a regularization technique aimed to prevent the appearance of *overfitting* [2]. This regularization is not necessarily present but let to the analyst. There are 5 major types of training algorithms:

- **Gradient descent:** it follows the direction of the negative gradient of the loss function, a learning rate is present as a parameter controlled to avoid a too pronounced update of the weights between iterations. It can be fixed or tuned for each iteration, but this algorithm usually requires a high number of iterations to reach convergence.

- **Newton's method:** uses a Hessian matrix which inverse premultiplies the gradient, and a fixed learning rate. The weight updating rule is very similar to the previous case.

- **Conjugate gradient:** it is half-way between previous algorithms. Considerably faster than gradient descent, performs a good choice for large networks as it does not include the Hessian (neither its inversion) what speeds up the updating process considerably. Weights are updated in correspondence with a training direction vector, originally set to be the negative gradient direction.

- **Quasi-Newton method:** this method works in the same way as Newton's method, but dispenses with the Hessian matrix. Another matrix is used instead, and approximated from loss function's first derivatives at each iteration. Different methods are available to find this matrix (BFGS [3] is the most extended).

- **Levenberg-Marquardt algorithm:** it is designed to deal with loss functions that resemble a sum of squared errors, something quite practical in many engineering applications. A Jacobian matrix of the errors is computed and then used to determine the loss gradient. The weight updating is performed by using the Jacobian matrix together with a damping parameter to control the updating. The parameter measures how far the method is from Newton's method (0) or gradient descent (1). The algorithm is very fast for the specific class of loss functions but can not be employed with other loss metrics. Its performance is also impoverished when using large size networks.

[2] Excessive performance of the model in the training set, difficulting its capacity to generalize for cases out of the training dataset. It is typically observed as an overestimation of the performance in a *validation* set over that in a *training* set.

[3] Broyden-Fletcher-Glodfarb-Shanno formula

An addition to the previous comes in the form of the *Extreme Learning Machine* algorithm, detailed in Section [4]. The training process is extremely fast provided it uses a shallow layout compared to other deeper architectures, in the form of a *Single Layer Feedforward Network* with only one hidden layer of neurons. When used in the context of PINN, this is particularly significant and the solution can be purely driven by the fulfillment of the physics of the problem.

In a broad sense, Machine Learning and more specifically ANNs introduce a series of advantages over other solution strategies. In particular, their capacity to generalize to previously unseen data and the fast evaluation of predictions once trained make them suitable to perform large processes of design iterations. Additionally, obtained predictions can be used to form a dataset that feeds a more complex network in order to recover other kinds of solutions such as non-linear behaviours. On the other hand, the training process can be very slow, specially when large datasets are required for training or deep networks are used, as the number of connections increases rapidly. Moreover, the selection of certain parameters such as the number of hidden nodes or the specification of the learning ratio can affect the quality of the solution and require the network to be re-trained by trial and error in most cases.

3 The Sublaminate Generalized Unified Formulation: S-GUF

Material science and structural design seek for unified theories that could describe complex assemblies of different materials, shapes and other conditions. The appearance of new material concepts make it necessary for this formulation to be modular and able to introduce new terms that account for difficult behaviours while maintaining coherence. Moreover, their application to traditional materials such as metals should be kept up to date and be covered as well. The *Sublaminate Generalized Unified Formulation* combines all of these characteristics in an elegant way, with a compact indicial notation that can deal with any-kind expansion order and theory. Different kinematic refinements can be used in different subregions, obtaining multiple axiomatic 2D models with different kinematics. The stated formulation in this chapter is considered for both plates and cylindrical shells.

3.1 Notation and domain definition

SGUF uses a form of Einstein's notation to differentiate the contributions from different sublaminates, plies and kinematic variables. All the formulation is referred to a Cartesian frame, with some minor differences between plates and shells. The X direction is always referred in the longitudinal direction, Z direction goes in the thickness direction and Y direction is orthogonal to the previous two, to close the orthogonal system definition. In the context of shells, Y and Z directions are referred to the tangential and normal of the panel. An illustration of this particular reference system is given in Fig.(2):

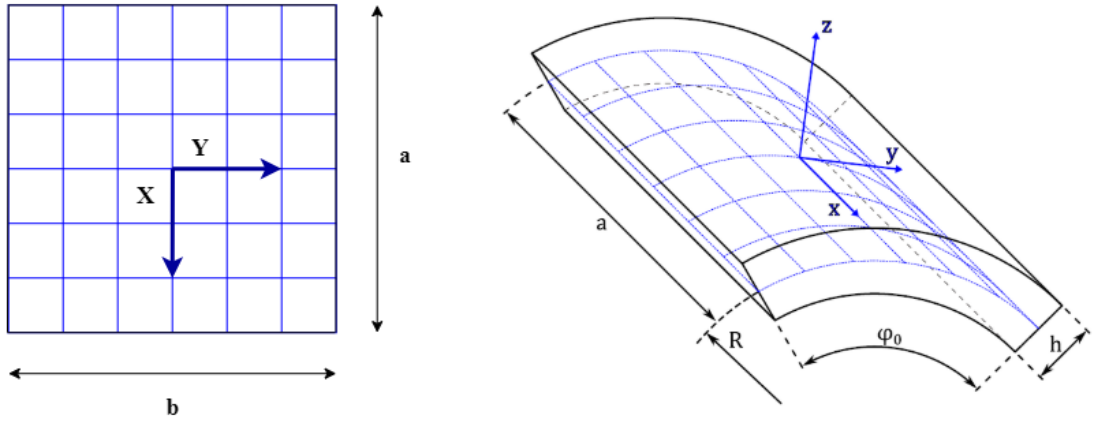


Figure 2: Plate diagram (*left*) seen from the top, and shell diagram (*right*).

The definition of the kinematic model is referred to the *midsurface*, that serves as origin for the thickness direction. A plate is defined by three characteristic lengths $\{a, b, h\}$, the first two planar quantities and the thickness, all referred to this surface. Shell definition requires of two additional quantities, the radius of curvature, R, and a subtended angle, ϕ, that make the transverse length taken to be the arclength, $b = R\phi$. Considered cylindrical shells are *opened shells* ($\phi < 360$) only, and the approach works for constant radii/constant thickness geometries. The domain of reference in both cases is defined as $x \in [-a/2, a/2]$, $y \in [-b/2, b/2]$ and $z \in [-h/2, h/2]$ (origin set in the middle), while the through-thickness direction could be spanned by multiple layers depending on the case under analysis. A set of local coordinates are also defined and referred to a ply or sublaminate from its thickness midpoint, $z_p \in [-h_p/2, h_p/2]$ and $z_k \in [-h_k/2, h_k/2]$, these are involved in the definition of through-thickness kernels and

recovery of solution fields at local level.

The presence of superscripts p, k refer to the index number of current ply and sublaminate. Sublaminates are referred in decreasing order from top to bottom, while plies follow the same order. ESL theories use the single k superscript as their treatment is independent on the ply layup. On the contrary LW theories require a specific inclusion of the ply index for further assembly. Two kinds of subscripts are present, the first is referred to the axes of reference to which a specific variable belongs, $r = x, y, z$. It is sometimes referred with a second index, $s = x, y, z$, when two sets of variables are conjugate, for instance, inside a variational statement. The other subscript is given by α_{u_r} (or analogously β_{u_s}) and referred to a specific kinematic variable term. It is meant to span $\alpha_{u_r} = 0...N_{u_r}^{p,k}$, where $N_{u_r}^{p,k}$ refers to the maximum expansion order given by the kinematic theory in a specific direction. Thus, a displacement component in ESL and LW theories depend on all coordinates, while kinematic variables rely only on the axiomatic definition, in-plane coordinates. This dependence is introduced in Eq.(1):

$$\begin{aligned} u_r^k(x, y, z_k, t) &\longrightarrow u_{r\alpha_{u_r}}^k(x, y, t) \\ u_r^{p,k}(x, y, z_p, t) &\longrightarrow u_{r\alpha_{u_r}}^{p,k}(x, y, t) \end{aligned} \tag{1}$$

Different kinematic theories are also reported using different abbreaviations. ESL theories are referred as $ED_{N_{u_x}(Z)N_{u_y}(Z)N_{u_z}(Z)}$ either for the whole plate or an individual sublaminate, where Z is only present if that displacement component is enhanced by a zig-zag description. For LW theories, they are abbreviated as $LD_{N_{ux}(E)N_{uy}(E)N_{uz}(E)}$, where E refers to that component described with an ESL model on a partially layer-wise theory, if present.

The *tilde* symbol in the context of the constitutive relation ($\tilde{C}$) refers to the rotation from the lamina to the laminate reference frame. Subscripts Ω and n are referred to the in-plane and transverse subsets, and can also appear as a dual combination $\Omega\Omega$,Ωn, $n\Omega$, nn, that state for the different partitions of the constitutive relation matrix.

3.2 Kinematic description

ESL models comprehend those that describe the behaviour of a whole sublaminate as of a single ply, without considering piece-wise property variations between different laminates. Its general expression for the displacements makes use of a series of through-thickness functions that describe the influence of each kinematic variable in the transverse direction:

$$\begin{aligned} u_x^k &= F_{\alpha_{ux}}^{x,k} u_{x_{\alpha_{ux}}}^k \\ u_y^k &= F_{\alpha_{uy}}^{y,k} u_{y_{\alpha_{uy}}}^k \\ u_z^k &= F_{\alpha_{uz}}^{z,k} u_{z_{\alpha_{uz}}}^k \end{aligned} \tag{2}$$

where $F_{\alpha_{ur}}^{r,k}$ (r=x,y,z) refers to the specific through-thickness function associated to $u_{r_{\alpha_{ur}}}^k$ kinematic variable. The sublaminate index makes this model independent on the ply count. It means the displacement field shape is smoothened and do not carry the inherent piecewise continuity that characterizes higher fidelity models. ESL theories can be enhanced with the addition of zig-zag terms that artificially implement a contribution, $F_{ZZ} = (-1)^p \zeta_p$, that varies its sign on each ply, increasing fidelity at lower cost. ζ_p states for a $[-1, 1]$ non-dimensionalization of local z_p coordinate.

LW models overcome some of their limitations to provide accurate solutions in highly anisotropic laminated composite panels or whenever there is an interest in the transverse component of 3D stresses. It deals with the increased complexity of the analysis by particularizing the degrees of freedom at ply level:

$$\begin{aligned} u_x^{p,k} &= F_{\alpha_{ux}}^{x,p,k} u_{x_{\alpha_{ux}}}^{p,k} \\ u_y^{p,k} &= F_{\alpha_{uy}}^{y,p,k} u_{y_{\alpha_{uy}}}^{p,k} \\ u_z^{p,k} &= F_{\alpha_{uz}}^{z,p,k} u_{z_{\alpha_{uz}}}^{p,k} \end{aligned} \tag{3}$$

where $F_{\alpha_{ur}}^{r,p,k}$ (r=x,y,z) refers to the specific through-thickness function associated to $u_{r_{\alpha_{ur}}}^{p,k}$ kinematic variable.

There is a variation of this description that could result of great interest in some cases. It consists on the partial definition of some displacements with an ESL theory, while the others are LW, or viceversa. It maintains the piecewise description for those displacements (improved accuracy) while the computational load is mildly alleviated in the directions that are not of biggest interest (see Eq.(4)). It is referred as *partially layer-wise* model (in this case applied to the transverse displacement component):

$$\begin{aligned} u_x^{p,k} &= F_{\alpha_{ux}}^{x,p,k} u_{x_{\alpha_{ux}}}^{p,k} \\ u_y^{p,k} &= F_{\alpha_{uy}}^{y,p,k} u_{y_{\alpha_{uy}}}^{p,k} \\ u_z^{k} &= F_{\alpha_{uz}}^{z,k} u_{z_{\alpha_{uz}}}^{k} \end{aligned} \tag{4}$$

The through-thickness functions are taken as *Legendre polynomials*, which has an inherent suitability to impose compatibility conditions across plies. These functions take a zero value except for the function associated to u_{r_0} or u_{r_1} when the displacement field is to be evaluated at the top or bottom of a subregion, respectively. In this way, the displacement component is uniquely defined by a single kinematic variable in those situations, and the interlayer compatibility can be easily enforced. There are some requirements that are mandatory whenever working with kinematic models in a multilayered plate framework:

- Displacement continuity (essential conditions) must be ensured at the interfaces between subregions, either plies or sublaminates.
- Natural conditions are present in terms of interlaminar equilibrium at ply interfaces for the transverse stress components.

The first condition is satisfied thanks to an *assembly procedure*, where equivalent degrees of freedom at the interfaces between plies and sublaminates are identified. As of the kinematic description, the first two kinematic variables u_{r_0}/u_{r_1} (whenever $N_{ur} > 1$) are present when two regions are connected, and their interface is twice described by equivalent kinematic variables. The assembly makes this description unique by eliminating the duplicity, what in other way makes the governing equations to be coupled, and the structure to behave in a compatible way fulfilling the interlaminar compatibility. Additionally, in the context of a variational statement, when this is expanded the associated variables undergo an *expansion process* in which the components compactly embedded in the notation are cycled, expressing their contributions explicitely. A good example of this expansion process can be found in Appendix A, regarding the expression of the PVW statement.

3.2.1 Multiple kinematics: an example of application

It has been mentioned that the use of ESL/LW theories becomes a trade-off in the context of multilayered plates, between solution fidelity for the field variables in the thickness direction and computational complexity. In real life applications, the type of modeling of composite structures is usually dominated by the complexity of the solution field expected for the stacked layup of plies. It is then easy to find a vast majority of situations where there is a clear distinction between regions of interest (*sublaminates*) in the thickness direction of the domain. The S-GUF theory becomes even more valuable in these situations, when it is possible to combine ESL/LW theories at the different sublaminates to obtain a remarkable fidelity, while limiting the computational cost of those stakups that can be determined with lower order/simpler theories.

An excellent example of this situation can be found for the case of a sandwich panel. The panel is differentiated by an *inner core*, typically corresponding to the biggest thickness and made of a soft material subjected to non-linearities that makes it necessary to employ a much more complex description. Both skins could be described in a similar way with a simpler model and be made of one or more plies altogether with different fiber orientations. The application to the design of a rocket fairing can be considered to be pretty illustrative. It is alike to describe this structure as made of three differentiated sublaminates: an *outer skin* subjected to external loads, a *honeycomb* core, and an *inner skin* providing stiffness to the walls to avoid large deflections that could lead to catastrophical failure.

Reported in Fig. (3), the outer skin is made of four layers, and will be subjected to a rather anisotropic loading condition regarding aerodynamic loads, pressure variation with altitude and aeroacoustic couplings, therefore a complex piece-wise variation is expected in the thickness direction, better described with a layerwise model. The displacement field at the soft core must be complex in the radial direction, thus a high order model is expected while it can be described with an ESL theory provided the core is a single piece (could be treated as a single ply). The inner skin on the other hand has the least severe impact on the performance of the analysis, so it can be described with a rather simple FSDT model, leveraging in a way the domain complexity. It must be mentioned that if stiffeners were to be considered, they would be attached to the inner skin and a more complex interaction would be expected due to the load introduction into them.

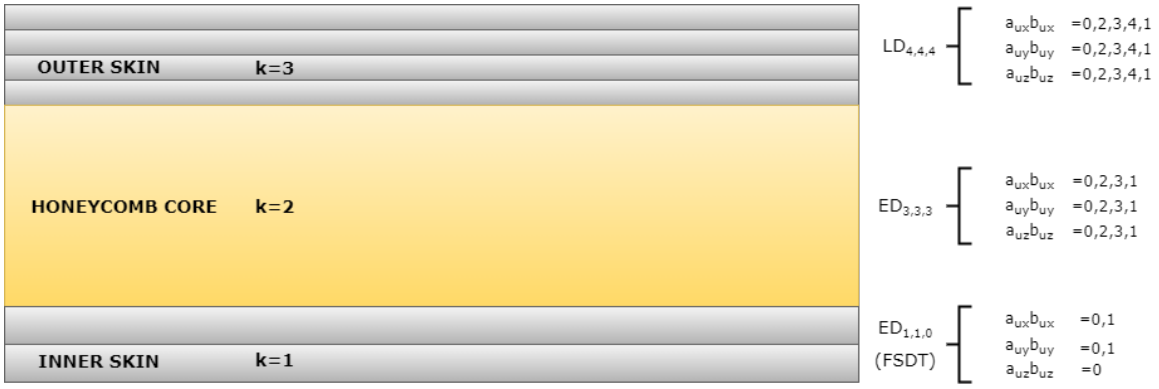


Figure 3: Shroud honeycomb structure with three differentiated regions.

Although this particular example of a sandwich panel, the complexity of the model at each sublaminate or the addition of new subregions with different materials is not limited. SGUF allows to introduce any kind of kinematic description along with the conditions to assemble the different sublaminate.

3.3 Material description

A 3D orthotropic description of the elasticity matrix is considered, but this choice is easily specified to handle isotropic materials. The latter assumption is consistent with the fact that typical materials in composite structures possess three ortogonal planes of elasticity. The law is modified according to matrix $\boldsymbol{T}$, that introduces a rotation from the lamina to the laminate reference frame, it is given in Eq.(5):

$$\tilde{\boldsymbol{C}} = \boldsymbol{T}\boldsymbol{C}\boldsymbol{T}' = \begin{bmatrix} \tilde{C_{11}} & \tilde{C_{12}} & \tilde{C_{13}} & 0 & 0 & \tilde{C_{16}} \\ \tilde{C_{12}} & \tilde{C_{22}} & \tilde{C_{23}} & 0 & 0 & \tilde{C_{26}} \\ \tilde{C_{13}} & \tilde{C_{23}} & \tilde{C_{33}} & 0 & 0 & \tilde{C_{36}} \\ 0 & 0 & 0 & \tilde{C_{44}} & \tilde{C_{45}} & 0 \\ 0 & 0 & 0 & \tilde{C_{45}} & \tilde{C_{55}} & 0 \\ \tilde{C_{16}} & \tilde{C_{26}} & \tilde{C_{36}} & 0 & 0 & \tilde{C_{66}} \end{bmatrix} \tag{5}$$

And the rotation is given by Eq.(6):

$$\boldsymbol{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 0 & 0 & 0 & -2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & 0 & 0 & 0 & 2\sin\theta\cos\theta \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ \sin\theta\cos\theta & -\sin\theta\cos\theta & 0 & 0 & 0 & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{6}$$

This transformation is remarkably important when dealing with multilayered panels, where their mechanical properties are influenced by the relative orientation of composite fibers inside the matrix. A general form of the constitutive law relating stresses and strains is obtained in Eq.(7):

$$\boldsymbol{\sigma} = \tilde{\boldsymbol{C}}\boldsymbol{\varepsilon} \tag{7}$$

Additionally, the constitutive matrix is partitioned into four block contributions that represent the purely in-plane and transverse components of the relation (diagonal blocks), and the coupled behaviour (off-diagonal blocks).

3.4 Stresses and strains derivation

Both strain and stress tensors in this formulation are distinguished in their *in-plane* and *transverse* (normal) contributions:

$$\begin{aligned} \boldsymbol{\sigma}_\Omega &= \tilde{C}_{\Omega\Omega}\boldsymbol{\varepsilon}_\Omega + \tilde{C}_{\Omega n}\boldsymbol{\varepsilon}_n \\ \boldsymbol{\sigma}_n &= \tilde{C}_{n\Omega}\boldsymbol{\varepsilon}_\Omega + \tilde{C}_{nn}\boldsymbol{\varepsilon}_n \end{aligned} \tag{8}$$

where $\boldsymbol{\varepsilon}_\Omega$, $\boldsymbol{\varepsilon}_n$ are subsets of the strain vector:

$$\begin{aligned} \boldsymbol{\varepsilon}_\Omega &= \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \gamma_{xy} \end{bmatrix}^T \\ \boldsymbol{\varepsilon}_n &= \begin{bmatrix} \gamma_{yz} & \gamma_{xz} & \varepsilon_{zz} \end{bmatrix}^T \end{aligned} \tag{9}$$

and $\boldsymbol{\sigma}_\Omega$, $\boldsymbol{\sigma}_n$ stand for the stresses counterpart:

$$\boldsymbol{\sigma}_\Omega = \begin{bmatrix}\sigma_{xx} & \sigma_{yy} & \tau_{xy}\end{bmatrix}^T$$
$$\boldsymbol{\sigma}_n = \begin{bmatrix}\tau_{yz} & \tau_{xz} & \sigma_{zz}\end{bmatrix}^T \tag{10}$$

$\tilde{C}_{\Omega\Omega}, \tilde{C}_{\Omega n}, \tilde{C}_{n\Omega}$ and $\tilde{C}_{nn}$ represent the different submatrices of the constitutive relation (see Appendix A). γ_{rs} and τ_{rs} are referred as engineering quantities introduced by the form of the constitutive relation, and equivalent to $2\varepsilon_{rs}$ and $2\sigma_{rs}$, respectively. This description is common to both plates and shells, with some differences on the derivation of strains. For a plate, the vector of strains is given by a matrix of derivatives that multiplies the vector of displacements:

$$\boldsymbol{\varepsilon} = \boldsymbol{D}\boldsymbol{u}$$
$$\boldsymbol{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & 0 & \frac{\partial}{\partial z} \end{bmatrix} \tag{11}$$

and is also particularized in in-plane and transverse contributions:

$$\boldsymbol{\varepsilon}_\Omega^{p,k} = \boldsymbol{D}_\Omega \boldsymbol{u}^{p,k}$$
$$\boldsymbol{\varepsilon}_n^{p,k} = \boldsymbol{D}_n \boldsymbol{u}^{p,k} + \boldsymbol{D}_z \boldsymbol{u}^{p,k} \tag{12}$$

where $\boldsymbol{D}_\Omega$, $\boldsymbol{D}_n$ and $\boldsymbol{D}_z$ are the corresponding submatrices in $\boldsymbol{D}$. The vector of displacements is given by the three Cartesian components and defined as $\boldsymbol{u}^{p,k} = [u_x^{p,k} \quad u_y^{p,k} \quad u_z^{p,k}]$.

In the case of shells, a general expression for a 3D strain definition is given for a general orthogonal frame, where displacements are identified as (u_1, u_2, u_3), in Eq.(13):

$$\begin{aligned}
\varepsilon_{11} &= \frac{1}{A_1}\left(\frac{\partial u_1}{\partial \xi_1} + \frac{1}{a_2}\frac{\partial a_1}{\partial \xi_2}u_2 + \frac{a_1}{R_1}u_3\right) \\
\varepsilon_{22} &= \frac{1}{A_2}\left(\frac{\partial u_2}{\partial \xi_2} + \frac{1}{a_1}\frac{\partial a_2}{\partial \xi_1}u_1 + \frac{a_2}{R_2}u_3\right) \\
\gamma_{12} &= \frac{A_2}{A_1}\frac{\partial}{\partial \xi_1}(u_2/A_2) + \frac{A_1}{A_2}\frac{\partial}{\partial \xi_2}(u_1/A_1) \\
\gamma_{23} &= \frac{1}{A_2}\frac{\partial u_3}{\partial \xi_2} + A_2\frac{\partial}{\partial \xi_3}(u_2/A_2) \\
\gamma_{13} &= \frac{1}{A_1}\frac{\partial u_3}{\partial \xi_1} + A_1\frac{\partial}{\partial \xi_3}(u_1/A_1) \\
\varepsilon_{33} &= \frac{\partial u_3}{\partial \xi_3}
\end{aligned} \tag{13}$$

where a_i states for the square-rooted diagonal components of the surface metric tensor [4], that take therefore the unit value. A_i are the *Lamé coefficients*, defined accordingly:

$$A_i = a_i\left(1 + \frac{\xi_3}{R_i}\right) \tag{14}$$

[4]Second order tensor used to define the shell surface in Euclidean space, in order to distingish it from a generalized curvilinear space.

These parameters hold in a strain-stress relationship as material dependent quantities that dictate Hooke's law. As a matter of fact, they are present at this point to represent that their relation with stresses is altered and dependent on the radius of curvature. R_i is meant as the radius in two directions, however, the simplification from cylindrical shells states that $R_1 = R_x = \infty$ and $R_2 = R_y = R$, thus, a single radius of curvature is taken into account.

By changing the previous general frame into the given Cartesian frame of reference, the strains are turned into a much familiar form:

$$\begin{aligned}
\varepsilon_{xx} &= \frac{\partial u_x}{\partial x} \\
\varepsilon_{yy} &= \frac{1}{1+z/R}(\frac{\partial u_y}{\partial y} + \frac{u_z}{R}) \\
\gamma_{xy} &= \frac{\partial u_y}{\partial x} + \frac{1}{1+z/R}\frac{\partial u_x}{\partial y} \\
\gamma_{yz} &= \frac{\partial u_y}{\partial z} + \frac{1}{1+z/R}(\frac{\partial u_z}{\partial y} - \frac{u_y}{R}) \\
\gamma_{xz} &= \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \\
\varepsilon_{zz} &= \frac{\partial u_z}{\partial z}
\end{aligned} \tag{15}$$

The presence of the curvature in the denominator of every normal or shear component that includes the tangential axis direction is coherent with the fact that when $\frac{1}{R} \longrightarrow 0$, these expressions progressively turn into those of a plate. The compact vectorial form of these expressions turns into:

$$\begin{cases}
\boldsymbol{\varepsilon_\Omega}^{p,k} = (\boldsymbol{D}_\Omega + \boldsymbol{A}_\Omega)\boldsymbol{u}^{p,k} \\
\boldsymbol{\varepsilon_n}^{p,k} = (\boldsymbol{D}_n - \boldsymbol{A}_n + \boldsymbol{D}_z)\boldsymbol{u}^{p,k}
\end{cases} \tag{16}$$

where $\boldsymbol{D}_\Omega$, $\boldsymbol{A}_\Omega$, $\boldsymbol{D}_n$, $\boldsymbol{A}_n$ and $\boldsymbol{D}_z$ are the compact matricial form of the components accompaniying the derivatives in Eq.(15).

3.5 Problem formulation

Two different approaches are derived based on SGUF theory. A *strong-form formulation* is obtained from the derivation of a system of partial differential equations from the variational statement of the *Principle of Virtual Works* (PVW) in the first place. Additionally, a *weak-form formulation* is obtained using the total potential energy, $\Pi = U + V$, as a functional to be minimized through a set of stationarity conditions. The integral forms of strain energy, U, and the work done by external forces, V, are obtained from the PVW where integrals are evaluated numerically.

The difference with *weak form* equations is on the treatment of the approximation. While weak form formulations make use of approximate representations of the displacement fields that are directly introduced into the variational statement, strong form treats the integration by parts of the equations and looks for a suitable manner to impose them in an exact way. The kinematic components of the model for every displacement in a SGUF theory are referred indistinctly as *kinematic variables* or *degrees of freedom* in the following for simplicity.

3.5.1 Strong-form formulation

The *Principle of Virtual Works* (PVW) states that an elastic structure is considered in equilibrium under a certain load if for any virtual variation of a displacement component from a compatible state of deformation, the virtual work equals the virtual strain energy. This can be seen as an a priori state of equivalence between external work and internal (deformation) work as reported in Eq.(17). The statement of the PVW can be recovered from a three dimensional definition:

$$\begin{gathered} \delta W_i = \delta W_e \\ \iiint_V \delta \boldsymbol{\varepsilon}^{\boldsymbol{T}} \boldsymbol{\sigma} dV = - \iiint_V \delta \boldsymbol{u}^{\boldsymbol{T}} \rho \ddot{\boldsymbol{u}} dV + \int_\Omega \delta \boldsymbol{u} \boldsymbol{f} \end{gathered} \tag{17}$$

where $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ correspond to the strain and stress tensors, respectively. f states for a surface load in this particular example, but any other external load could be considered. V is the volume, and Ω introduces the surface integral, while ρ holds for the material density. It is worth mentioning the second term of Eq.(17) represents the dynamics of the structure. It plays an important role for the *free vibrations* problem where the target is to find those first natural frequencies that excite vibration modes on the plate, a typical requirement when designing a structure. The volume integration of the density is seen as an equivalent to the mass of the plate in the PVW.

A series of *kernels* are defined in Eq.(18) to represent the transverse description embedded by the through-thickness functions inside the PVW statement, and particular to the case of plates:

$$Z^{p\alpha_{ur}\beta_{us}}_{(\partial)u_r(\partial)u_s} = \int_{z_p^{bot}}^{z_p^{top}} F_{\alpha_{ur}(,z)} F_{\beta_{us}(,z)} dz \qquad (r,s = x,y,z) \tag{18}$$

where the limits of integration are referred to the top and bottom faces of a generic ply. The ∂ symbol refers to the partial differentiation of through-thickness functions with respect to the transverse direction, something that becomes relevant for some terms in the expanded form of the PVW. In the context of shells, the kernels are defined in a different way, due to the presence of a curvature:

$$Z^{p\alpha_{ur}\beta_{us}}_{(\partial)u_r(\partial)u_s dn} = \int_{z_p^{bot}}^{z_p^{top}} \frac{R^n}{(R+z)^d} F_{\alpha_{ur}(,z)} F_{\beta_{us}(,z)} (1 + \frac{z}{R}) dz \qquad (r,s = x,y,z) \quad (d,n = 0,1,2) \tag{19}$$

where d,n are referred as additional terms related to the radius and transverse component. ∂ symbol holds with the same meaning as for plates. An expanded form of the complete PVW statement is given in Appendix A, expressed with SGUF indicial notation for both plates and shells. By applying integration by parts over that lenghty expression of the expanded form of PVW, it is possible to explicit all contributions in brackets corresponding to every virtual displacements. Eq.(20) shows an example of the application of this integration rule for the first term, corresponding to the variational displacement, δu_x:

$$\int_\Omega \frac{\partial u^{p,k}_{x\alpha_{ux}}}{\partial x}[\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial u^{p,k}_{x\beta ux}}{\partial x} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial u^{p,k}_{y\beta uy}}{\partial y} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial u^{p,k}_{x\beta ux}}{\partial y}$$
$$+ \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial u^{p,k}_{y\beta uy}}{\partial x}]d\Omega = \int_\Gamma \frac{\partial u^{p,k}_{x\alpha_{ux}}}{\partial x}[\boxed{\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial u^{p,k}_{x\beta ux}}{\partial x}} + \boxed{\tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial u^{p,k}_{y\beta uy}}{\partial y}}$$
$$+ \boxed{\tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial u^{p,k}_{x\beta ux}}{\partial y}} + \boxed{\tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial u^{p,k}_{y\beta uy}}{\partial x}}]d\Gamma - \int_\Omega \partial u^{p,k}_{x\alpha_{ux}}[\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial^2 u^{p,k}_{x\beta ux}}{\partial x^2}$$
$$+ \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial^2 u^{p,k}_{y\beta uy}}{\partial y \partial x} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{uxux} \frac{\partial^2 u^{p,k}_{x\beta ux}}{\partial y \partial x} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{uxuy} \frac{\partial^2 u^{p,k}_{y\beta uy}}{\partial x^2}]d\Omega \tag{20}$$

where Ω represents the surface in the domain, and Γ represents a one-dimensional surface (edge), where the corresponding boundary conditions are applied. Extracting the terms under Ω and Γ integrals, the governing equations and boundary conditions (boxed items contributions) are obtained respectively in the form of a system of partial differential equations. In this way, a system with as many equations as kinematic variables per index p, k (depending on whether ESL/LW theories are applied) is reported for each principal direction in its Cartesian frame. A general form of this system of equations is given by Eq.(21)-(23):

$$\begin{aligned}
&\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},xx} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},yx} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},yx} \\
&+ \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},xx} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},xy} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},yy} \\
&+ \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},yy} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},xy} - \tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_x u_z} u^{p,k}_{z\beta_{uz},x} \\
&- \tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{ux}}_{\partial u_x \partial u_x} u^{p,k}_{x\beta_{ux}} - \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_x u_z} u^{p,k}_{z\beta_{uz},y} - \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_x \partial u_y} u^{p,k}_{y\beta_{uy}} \\
&+ \tilde{C}^{p,k}_{13} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z} u^{p,k}_{z\beta_{uz},y} = f^{p,k}_x \qquad \forall p,k
\end{aligned} \tag{21}$$

$$\begin{aligned}
&\tilde{C}^{p,k}_{12} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},xy} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},yy} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},yy} \\
&+ \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},xy} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},xx} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},yx} \\
&+ \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},yx} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},xx} - \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_y u_z} u^{p,k}_{z\beta_{uz},x} \\
&- \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{ux}}_{\partial u_y \partial u_x} u^{p,k}_{x\beta_{ux}} - \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_y u_z} u^{p,k}_{z\beta_{uz},y} - \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_y \partial u_y} u^{p,k}_{y\beta_{uy}} \\
&+ \tilde{C}^{p,k}_{23} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z} u^{p,k}_{z\beta_{uz},y} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z} u^{p,k}_{z\beta_{uz},x} = f^{p,k}_y \qquad \forall p,k
\end{aligned} \tag{22}$$

$$\begin{aligned}
&\tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},xx} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x} u^{p,k}_{x\beta_{uy},x} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},yx} \\
&+ \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y} u^{p,k}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},xy} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x} u^{p,k}_{x\beta_{ux},y} \\
&+ \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},yy} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y} u^{p,k}_{y\beta_{uy},y} - \tilde{C}^{p,k}_{13} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_z u_x} u^{p,k}_{x\beta_{ux},x} \\
&- \tilde{C}^{p,k}_{23} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_z \partial u_y} u^{p,k}_{y\beta_{uy},y} - \tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_z u_x} u^{p,k}_{x\beta_{ux},y} - \tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_z \partial u_y} u^{p,k}_{y\beta_{uy},x} \\
&- \tilde{C}^{p,k}_{33} Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z \partial u_z} u^{p,k}_{z\beta_{uz}} = f^{p,k}_z \qquad \forall p,k
\end{aligned} \tag{23}$$

Moreover, the number of equations is sligthly reduced during the assembly procedure due to the fulfillment of the interlayer compatibility, in a similar manner of what is shown in Section [3.2]. Given a certain model order with n kinematic variables, the same number of equations

is generated, with the same indexing. This means there will be a given equation (except for $N_{u_r} = 0$) corresponding to variation $\delta u_{r1}/\delta u_{r0}$, that completely describes the physical behaviour of the ply at its top and bottom on its own, respectively. The particular difference of this approach with previously introduced assembly has to do with the interface between two connected regions, that is then defined by two equations representing the same information, they are equivalent and one is discarded from the problem to make it consistent (equal number of equations and unknowns). This procedure is the same performed over kinematic variables, what makes the system to be fully coupled and the output displacements to be compatible. It is performed both at ply and sublaminate level, and not procuring it leads to a solution which is not respectful of the interply compatibility requirements.

Natural conditions can be enforced for each degree of freedom, depending on the problem at hand, grouping the terms under the line integrals Γ:

$$\begin{cases} \tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},y} \\ +\tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{13} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z} u^{p,k}_{z\beta_{uz}} = 0 \\ \\ \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux},y} \\ +\tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y} u^{p,k}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z} u^{p,k}_{z\beta_{uz}} = 0 \qquad \forall p,k \end{cases} \tag{24}$$

$$\begin{cases} \tilde{C}^{p,k}_{12} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},y} \\ +\tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z} u^{p,k}_{z\beta_{uz}} = 0 \\ \\ \tilde{C}^{p,k}_{16} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x} u^{p,k}_{x\beta_{ux},y} \\ +\tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z} u^{p,k}_{z\beta_{uz}} = 0 \qquad \forall p,k \end{cases} \tag{25}$$

$$\begin{cases} \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x} u^{p,k}_{x\beta_{uy}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},y} \\ +\tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y} u^{p,k}_{y\beta_{uy}} = 0 \\ \\ \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x} u^{p,k}_{x\beta_{ux}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz},y} \\ +\tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y} u^{p,k}_{y\beta_{uy}} = 0 \qquad \forall p,k \end{cases} \tag{26}$$

these conditions hold at the boundary regions. *Essential conditions* prevent one or another displacement by restricting a certain kinematic variable to a fixed value (i.e. $u^{p,k}_{r\alpha_{ur}} = 0$) at the points corresponding to the physical external boundaries. In the same manner, displacements different from zero can be imposed instead of forces or moments, a very similar feature to the capabilities offered by FEM software.

By attending to (24)-(26), it is seen they resemble the expressions of *generalized forces* and *moments*, which have to be in equilibrium with the external loads. By considering the simplest theory available (single-ply, Mindlin theory), it is possible to conceptually simplify these conditions to:

$$\begin{cases} N_x = 0 \\ N_{xy} = 0 \\ M_x = 0 \\ M_{xy} = 0 \end{cases} \qquad \begin{cases} N_y = 0 \\ N_{yx} = 0 \\ M_y = 0 \\ M_{yx} = 0 \end{cases} \qquad \begin{cases} Q_x = 0 \\ Q_y = 0 \end{cases} \tag{27}$$

They are in correspondence to the generalized quantities that can be found in the context of a Reissner-Mindlin plate, what exemplifies the physical intuition of this method. The equations in (24) correspond in this case to four equations rather than two when they are expanded, and the kinematic variables are cycled. With δu_{x0}, membrane conditions $\{N_x, N_{xy}\}$ are reported from the expression, while δu_{x1} makes a different couple for associated moments, $\{M_x, M_{xy}\}$. The same happens for (25), while (26) reduces to the two conditions on the shearing, $\{Q_x, Q_y\}$, typically present at free edges. The index notation serves to condense a wide range of conditions for many different cases dependent on the theory at hand, what makes this formulation elegant and compact. This equivalence to generalized quantities is not unique and only applies softly to this situation (FSDT relies on a 2D reduced constitutive law, which does not hold in this case) serving as an example of the physical meaning of the system, it would be different with a distinct theory and the significance of the boundary conditions would become more intricate to follow at higher orders. The great benefit of this formulation belongs to its capacity to extend the problem to higher order theories by simply assembling more contributions in the form of more equations and adding degrees of freedom into the problem. A number of conditions equal to the number of degrees of freedom must be enforced on each boundary region, either natural or essential ones.

This formulation has been indicated to rely on a *collocation method*. This means the equations are enforced at discrete points throughout the domain (both inner and boundary points) and once the system is solved, the solution can be extended to any other point, obtaining a solution field for the displacements. Second order quantities are then recovered by reapplying the deformation field derivation and constitutive relations.

3.5.2 Weak-form formulation: an energy method

An alternative way to formulate the problem is to work directly with the variational statement. In the context of the PVW, it means dealing with the integral terms in an approximated way, for this purpose, the discretization of the integrand to be added in a summation term has been chosen as part of the method. The concept of this *numerical integration* is applicable for a discrete collocation problem, and is meant to avoid the symbolic integration of a mathematical expression over a certain domain in space, but rather to evaluate that expression at certain points and multiply by a differential of the domain that is assigned to that evaluation. From a geometrical perspective, it is equivalent to discretizing the domain of interest into small regions that could represent a surface ($\int_\Omega$) or edge portion ($\int_\Gamma$), and evaluate the inner integrand at the center of this region, as shown in Fig.(4). The summation of all these contributions is then the output of the integral, and the accuracy of the evaluation is given by the number of points (*grid size*) used for this purpose. A finer grid converges towards the exact result because the associated region of interest for a particular point evaluation aproaches a differential nature, in other words, the area under the curve represented by the integrand is covered more precisely. The domain must be properly sampled (including boundaries) in order to gain an accurate representation of the field variables. Low discrepancy sampling (relatively equispaced) is meant to cover adequately domains that typically will be rectangular or of any-other regular shape, thus a *uniform sampling* is considered default within this approach.

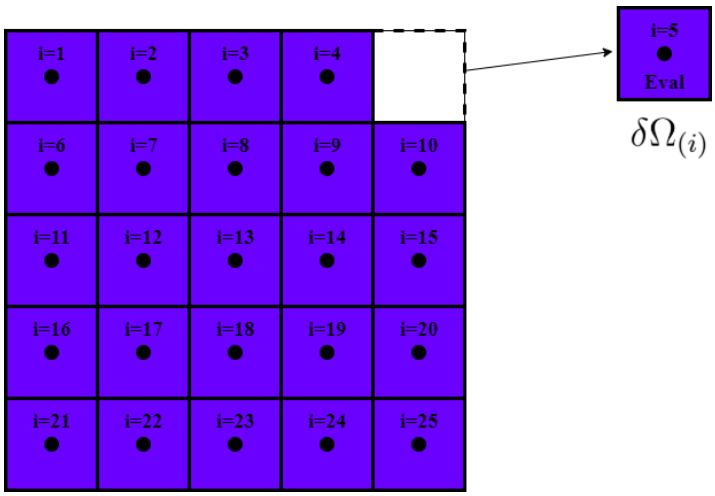


Figure 4: Example of domain subdivision within *numerical integration* for an arbitrary squared domain with $N = 5x5$ sample points. Function integrand is evaluated at the center point of each portion, and the coloured area bounds its applicability.

A simple description of the strain energy, U, can be given in the context of a Reissner-Mindlin plate in its integral form in Eq.(28):

$$U = \int_{\Omega} \frac{1}{2}(N_{\alpha\beta}\varepsilon^{0}_{\alpha\beta} + M_{\alpha\beta}\kappa_{\alpha\beta} + Q_{\alpha 3}\gamma_{\alpha 3})\delta\Omega \tag{28}$$

From a mathematical perspective, the evaluation of the *strain energy* via numerical integration can be expressed in the following way:

$$\begin{aligned} \bar{U}^{(i)} &= \frac{1}{2}(N^{(i)}_{\alpha\beta}\varepsilon^{0(i)}_{\alpha\beta} + M^{(i)}_{\alpha\beta}\kappa^{(i)}_{\alpha\beta} + Q^{(i)}_{\alpha 3}\gamma^{(i)}_{\alpha 3}) \\ U &= \sum_{i=1}^{N} \bar{U}^{(i)}\delta\Omega_{(i)} \qquad i = 1...N \quad (N >> 1) \end{aligned} \tag{29}$$

where N is the number of sample points used in the discretization and *bar* quantities indicate the evaluation of the integrand at the center of each small region. This same procedure is extended for the SGUF formulation, with a rather more complex expression of the strain energy obtained from the PVW statement, while the numerical integration is applied in the very same way.

In this context, the *total potential energy* is treated as a cost function which has to be minimized, defined as:

$$\Pi = U + V + T \tag{30}$$

stationarity conditions are imposed to drive the value of Π to a stationary point where an infinitesimal variation of the independent variables is meant to involve no change in energy. In the context of SGUF, these conditions are applied over every degree of freedom:

$$\frac{\partial \Pi}{\partial u^{p,k}_{r\alpha_{ur}}} = 0 \qquad (r = x, y, z) \quad (\alpha_{ur} = 1...N^{p,k}_{u_r}) \tag{31}$$

Internal (U) and external (V) energies can be recovered from the internal and external virtual works (variation of energy) present in PVW. The ultimate term, T, is a *penalty contribution* that introduces a set of constraints from an energy viewpoint of view, the *essential boundary*

conditions. Natural conditions are *a priori* enforced and need not be imposed, what saves the integration by parts performed for the strong approach. The recovery of all contributions from the expanded form of the PVW is directly taken from the weak statement, with no need to apply prior integration by parts. Eq.(32) shows the final expression for the strain energy, applicable to any kind of multilayered plate:

$$
\begin{aligned}
U = \sum_{i=1}^{Q_p} \bar{U}^{(i)} \delta\Omega_{(i)} = \\
\sum_{i=1}^{Q_p} \sum_{k=1}^{N_k} \sum_{p=1}^{N_p^k} \{ & u_{x\alpha_{ux},x}^{p,k,(i)} [\tilde{C}_{11}^{p,k} Z_{u_x u_x}^{p\alpha_{ux}\beta_{ux}} u_{x\beta_{ux},x}^{p,k,(i)} + \tilde{C}_{12}^{p,k} Z_{u_x u_y}^{p\alpha_{ux}\beta_{uy}} u_{y\beta_{uy},y}^{p,k,(i)} + \tilde{C}_{16}^{p,k} Z_{u_x u_x}^{p\alpha_{ux}\beta_{ux}} u_{x\beta_{ux},y}^{p,k,(i)} \\
& + \tilde{C}_{16}^{p,k} Z_{u_x u_y}^{p\alpha_{ux}\beta_{uy}} u_{y\beta_{uy},x}^{p,k,(i)}] + u_{y\alpha_{uy},y}^{p,k,(i)} [\tilde{C}_{12}^{p,k} Z_{u_y u_x}^{p\alpha_{uy}\beta_{ux}} u_{x\beta_{ux},x}^{p,k,(i)} + \tilde{C}_{22}^{p,k} Z_{u_y u_y}^{p\alpha_{uy}\beta_{uy}} u_{y\beta_{uy},y}^{p,k,(i)} \\
& + \tilde{C}_{26}^{p,k} Z_{u_y u_x}^{p\alpha_{uy}\beta_{ux}} u_{x\beta_{ux},y}^{p,k,(i)} + \tilde{C}_{26}^{p,k} Z_{u_y u_y}^{p\alpha_{uy}\beta_{uy}} u_{y\beta_{uy},x}^{p,k,(i)}] + u_{x\alpha_{ux},y}^{p,k,(i)} [\tilde{C}_{16}^{p,k} Z_{u_x u_x}^{p\alpha_{ux}\beta_{ux}} u_{x\beta_{ux},x}^{p,k,(i)} \\
& + \tilde{C}_{26}^{p,k} Z_{u_x u_y}^{p\alpha_{ux}\beta_{uy}} u_{y\beta_{uy},y}^{p,k,(i)} + \tilde{C}_{66}^{p,k} Z_{u_x u_x}^{p\alpha_{ux}\beta_{ux}} u_{x\beta_{ux},y}^{p,k,(i)} + \tilde{C}_{66}^{p,k} Z_{u_x u_y}^{p\alpha_{ux}\beta_{uy}} u_{y\beta_{uy},x}^{p,k,(i)}] \\
& + u_{y\alpha_{uy},x}^{p,k,(i)} [\tilde{C}_{16}^{p,k} Z_{u_y u_x}^{p\alpha_{uy}\beta_{ux}} u_{x\beta_{ux},x}^{p,k,(i)} + \tilde{C}_{26}^{p,k} Z_{u_y u_y}^{p\alpha_{uy}\beta_{uy}} u_{y\beta_{uy},y}^{p,k,(i)} + \tilde{C}_{66}^{p,k} Z_{u_y u_x}^{p\alpha_{uy}\beta_{ux}} u_{x\beta_{ux},y}^{p,k,(i)} \\
& + \tilde{C}_{66}^{p,k} Z_{u_y u_y}^{p\alpha_{uy}\beta_{uy}} u_{y\beta_{uy},x}^{p,k,(i)}] + u_{z\alpha_{uz},x}^{p,k,(i)} [\tilde{C}_{55}^{p,k} Z_{u_z u_z}^{p\alpha_{uz}\beta_{uz}} u_{z\beta_{uz},x}^{p,k,(i)} + \tilde{C}_{55}^{p,k} Z_{u_z \partial u_x}^{p\alpha_{uz}\beta_{ux}} u_{x\beta_{ux}}^{p,k,(i)} \\
& + \tilde{C}_{45}^{p,k} Z_{u_z u_z}^{p\alpha_{uz}\beta_{uz}} u_{z\beta_{uz},y}^{p,k,(i)} + \tilde{C}_{45}^{p,k} Z_{u_z \partial u_y}^{p\alpha_{uz}\beta_{uy}} u_{y\beta_{uy}}^{p,k,(i)}] + u_{z\alpha_{uz},y}^{p,k,(i)} [\tilde{C}_{45}^{p,k} Z_{u_z u_z}^{p\alpha_{uz}\beta_{uz}} u_{z\beta_{uz},x}^{p,k,(i)} \\
& + \tilde{C}_{45}^{p,k} Z_{u_z \partial u_x}^{p\alpha_{uz}\beta_{ux}} u_{x\beta_{ux}}^{p,k,(i)} + \tilde{C}_{44}^{p,k} Z_{u_z u_z}^{p\alpha_{uz}\beta_{uz}} u_{z\beta_{uz},y}^{p,k,(i)} + \tilde{C}_{44}^{p,k} Z_{u_z \partial u_y}^{p\alpha_{uz}\beta_{uy}} u_{y\beta_{uy}}^{p,k,(i)}] \\
& + u_{x\alpha_{ux}}^{p,k,(i)} [\tilde{C}_{55}^{p,k} Z_{\partial u_x u_z}^{p\alpha_{ux}\beta_{uz}} u_{z\beta_{uz},x}^{p,k,(i)} + \tilde{C}_{55}^{p,k} Z_{\partial u_x \partial u_x}^{p\alpha_{ux}\beta_{ux}} u_{x\beta_{ux}}^{p,k,(i)} + \tilde{C}_{45}^{p,k} Z_{\partial u_x u_z}^{p\alpha_{ux}\beta_{uz}} u_{z\beta_{uz},y}^{p,k,(i)} \\
& + \tilde{C}_{45}^{p,k} Z_{\partial u_x \partial u_y}^{p\alpha_{ux}\beta_{uy}} u_{y\beta_{uy}}^{p,k,(i)}] + u_{y\alpha_{uy}}^{p,k,(i)} [\tilde{C}_{45}^{p,k} Z_{\partial u_y \partial u_z}^{p\alpha_{uy}\beta_{uz}} u_{z\beta_{uz},x}^{p,k,(i)} + \tilde{C}_{45}^{p,k} Z_{\partial u_y \partial u_x}^{p\alpha_{uy}\beta_{ux}} u_{x\beta_{ux}}^{p,k,(i)} \\
& + \tilde{C}_{44}^{p,k} Z_{\partial u_y u_z}^{p\alpha_{uy}\beta_{uz}} u_{z\beta_{uz},y}^{p,k,(i)} + \tilde{C}_{44}^{p,k} Z_{\partial u_y \partial u_y}^{p\alpha_{uy}\beta_{uy}} u_{y\beta_{uy}}^{p,k,(i)}] + u_{x\alpha_{ux},x}^{p,k,(i)} \tilde{C}_{13}^{p,k} Z_{u_x \partial u_z}^{p\alpha_{ux}\beta_{uz}} u_{z\beta_{uz}}^{p,k,(i)} \\
& + u_{y\alpha_{uy},y}^{p,k,(i)} \tilde{C}_{23}^{p,k} Z_{u_y \partial u_z}^{p\alpha_{ux}\beta_{uz}} u_{z\beta_{uz}}^{p,k,(i)} + u_{x\alpha_{ux},y}^{p,k,(i)} \tilde{C}_{36}^{p,k} Z_{u_x \partial u_z}^{p\alpha_{ux}\beta_{uz}} u_{z\beta_{uz}}^{p,k,(i)} + u_{y\alpha_{uy},x}^{p,k,(i)} \tilde{C}_{36}^{p,k} Z_{u_y \partial u_z}^{p\alpha_{uy}\beta_{uz}} u_{z\beta_{uz}}^{p,k,(i)} \\
& + u_{z\alpha_{uz}}^{p,k,(i)} \tilde{C}_{13}^{p,k} Z_{\partial u_z u_x}^{p\alpha_{uz}\beta_{ux}} u_{x\beta_{ux},x}^{p,k,(i)} + u_{z\alpha_{uz}}^{p,k,(i)} \tilde{C}_{23}^{p,k} Z_{\partial u_z u_y}^{p\alpha_{uz}\beta_{uy}} u_{y\beta_{uy},y}^{p,k,(i)} + u_{z\alpha_{uz}}^{p,k,(i)} \tilde{C}_{36}^{p,k} Z_{\partial u_z u_x}^{p\alpha_{uz}\beta_{ux}} u_{x\beta_{ux},y}^{p,k,(i)} \\
& + u_{z\alpha_{uz}}^{p,k,(i)} \tilde{C}_{36}^{p,k} Z_{\partial u_z u_y}^{p\alpha_{uz}\beta_{uy}} u_{y\beta_{uy},x}^{p,k,(i)} + u_{z\alpha_{uz}}^{p,k,(i)} \tilde{C}_{33}^{p,k} Z_{\partial u_z \partial u_z}^{p\alpha_{uz}\beta_{uz}} u_{z\beta_{uz}}^{p,k,(i)} \} \frac{\Omega_t}{Q_p}
\end{aligned}
\tag{32}
$$

where additionally to SGUF notation, i holds for the particular evaluation of the expression at a single point, Q_p for the number of sample points used in the discretization and Ω_t is the overall surface spanned by the domain (referred to the *midsurface*). In the same manner, the external energy contribution coming from externally applied loads has its energy counterpart easily recoverable from right-hand side of Eq.(17). This definition will be different depending on the kind of loading condition of the plate. Considering two uniform pressures acting at the top and bottom surfaces, the numerical integration is given by:

$$
\begin{aligned}
V = \sum_{j=1}^{Q_{bs}} \bar{V}^{(j)} \delta l_j = \\
\sum_{j=1}^{Q_{bs}} [u_{z0}^{N_p^k, N_k, (j)} f_z^{top} + u_{z1}^{1,1,(j)} f_z^{bot}] \frac{\Omega_t}{Q_p}
\end{aligned}
\tag{33}
$$

where j refers to the evaluation at a single point, and Q_{bs} refers to a subset of sample points from Q_p where the load is introduced. In this particular case, $Q_{bs} = Q_p$ as both loads are uniform pressures applied on all points. Furthermore, δl indicates a differential length (in this

case a surface), but when a line load is applied (i.e. *membrane force*) the fraction turns into $\frac{l_t}{Q_{bs}}$, where l_t represents the total length of an edge.

For the case of shells, additional contributions have to be added to the expression of the approximated strain energy in Eq.(34):

$$
\begin{aligned}
U = \sum_{i=1}^{Q_p} \bar{U}^{(i)} \delta\Omega_{(i)} = \\
\sum_{i=1}^{Q_p}\sum_{k=1}^{N_k}\sum_{p=1}^{N_p^k} \{ & u^{p,k,(i)}_{x\alpha_{ux},x}[\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x 00} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y 11} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{uz}\beta_{uz}}_{u_x u_z 10} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x 11} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y 00} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{13} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z 00} u^{p,k,(i)}_{z\beta_{uz}}] \\
& + u^{p,k,(i)}_{x\alpha_{ux},y}[\tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x 11} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y 22} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{ux}\beta_{uz}}_{u_x u_z 21} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x 22} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{uy}}_{u_x u_y 11} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{ux}\beta_{uz}}_{u_x \partial u_z 11} u^{p,k,(i)}_{z\beta_{uz}}] \\
& + u^{p,k,(i)}_{x\alpha_{ux}}[-\tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_x u_y 10} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_x \partial u_y 00} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_x u_z 11} u^{p,k,(i)}_{z\beta_{uz},y} \\
& + \tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{ux}}_{\partial u_x \partial u_x 00} u^{p,k,(i)}_{x\beta_{ux}} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_x u_z 00} u^{p,k,(i)}_{z\beta_{uz},x}] \\
& + u^{p,k,(i)}_{y\alpha_{uy},y}[\tilde{C}^{p,k}_{12} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x 11} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y 22} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uz}}_{u_y u_z 21} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x 22} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y 11} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z 11} u^{p,k,(i)}_{z\beta_{uz}}] \\
& + u^{p,k,(i)}_{y\alpha_{uy},x}[\tilde{C}^{p,k}_{16} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x 00} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y 11} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uz}}_{u_y u_z 10} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{ux}}_{u_y u_x 11} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y 00} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{uy}\beta_{uz}}_{u_y \partial u_z 00} u^{p,k,(i)}_{z\beta_{uz}}] \\
& + u^{p,k,(i)}_{y\alpha_{uy}}[\tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y 20} u^{p,k,(i)}_{y\beta_{uy}} - \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{u_y \partial u_y 10} u^{p,k,(i)}_{y\beta_{uy}} - \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uz}}_{u_y u_z 21} u^{p,k,(i)}_{z\beta_{uz},y} \\
& - \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{ux}}_{u_y \partial u_x 10} u^{p,k,(i)}_{x\beta_{ux}} - \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{uz}}_{u_y u_z 10} u^{p,k,(i)}_{z\beta_{uz},x}] \\
& + u^{p,k,(i)}_{y\alpha_{uy}}[-\tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_y u_y 10} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_y \partial u_y 00} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_y u_z 11} u^{p,k,(i)}_{z\beta_{uz},y} \\
& + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{ux}}_{\partial u_y \partial u_x 00} u^{p,k,(i)}_{x\beta_{ux}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_y u_z 00} u^{p,k,(i)}_{z\beta_{uz},x}] \\
& + u^{p,k,(i)}_{z\alpha_{uz}}[\tilde{C}^{p,k}_{12} Z^{p\alpha_{uz}\beta_{ux}}_{u_z u_x 10} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uz}\beta_{uy}}_{u_z u_y 21} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z 20} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_z u_x 21} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uz}\beta_{uy}}_{u_z u_y 10} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uz}\beta_{uz}}_{u_z \partial u_z 10} u^{p,k,(i)}_{z\beta_{uz}}] \\
& + u^{p,k,(i)}_{z\alpha_{uz},y}[-\tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uy}}_{u_z u_y 21} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y 11} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z 22} u^{p,k,(i)}_{z\beta_{uz},y} \\
& + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x 11} u^{p,k,(i)}_{x\beta_{ux}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z 11} u^{p,k,(i)}_{z\beta_{uz},x}] \\
& + u^{p,k,(i)}_{z\alpha_{uz},x}[-\tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uy}}_{u_z u_y 10} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uy}}_{u_z \partial u_y 00} u^{p,k,(i)}_{y\beta_{uy}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z u_z 11} u^{p,k,(i)}_{z\beta_{uz},y} \\
& + \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{ux}}_{u_z \partial u_x 00} u^{p,k,(i)}_{x\beta_{ux}} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z 00} u^{p,k,(i)}_{z\beta_{uz},x}] \\
& + u^{p,k,(i)}_{z\alpha_{uz}}[\tilde{C}^{p,k}_{13} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_z \partial u_x 00} u^{p,k,(i)}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_z u_y 11} u^{p,k,(i)}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z u_z 10} u^{p,k,(i)}_{z\beta_{uz}} \\
& + \tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_z u_x 11} u^{p,k,(i)}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_z u_y 00} u^{p,k,(i)}_{y\beta_{uy},x} + \tilde{C}^{p,k}_{33} Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z \partial u_z 00} u^{p,k,(i)}_{z\beta_{uz}}]\} \frac{\Omega_t}{Q_p}
\end{aligned}
\tag{34}
$$

In order to consider the free vibrations capability of the energy method, a similar treatment has been given to the inertial properties. The dynamic contribution from PVW is also numerically integrated as:

$$M = \sum_{i=1}^{Q_p} \bar{M}^{(i)} \delta\Omega_{(i)} =$$
$$\sum_{i=1}^{Q_p} \sum_{k=1}^{N_k} \sum_{p=1}^{N_p^k} [u_{x\alpha_{ux}}^{p,k} \rho^{p,k} Z_{u_x u_x(00)}^{p\alpha_{ux}\beta_{ux}} \ddot{u}_{x\beta_{ux}}^{p,k} + u_{y\alpha_{uy}}^{p,k} \rho^{p,k} Z_{u_y u_y(00)}^{p\alpha_{uy}\beta_{uy}} \ddot{u}_{y\beta_{uy}}^{p,k}$$
$$+ u_{z\alpha_{uz}}^{p,k} \rho^{p,k} Z_{u_z u_z(00)}^{p\alpha_{uz}\beta_{uz}} \ddot{u}_{z\beta_{uz}}^{p,k}] \frac{\Omega_t}{Q_p} \tag{35}$$

where the subscripts in brackets are not considered for plates. Physically, this can be understood as the equivalent to mass within this method. This *kinetic energy* contribution is therefore taken into account only when dealing with free vibration problems.

The penalty contribution in (30) is introduced whenever kinematic boundary conditions are present. Eq.(36) shows the penalty term defined for three kinematic boundary conditions:

$$T = \lambda_n \int_\Gamma |u_n - \hat{u}_n| \delta l + \lambda_t \int_\Gamma |u_t - \hat{u}_t| \delta l + \lambda \int_\Gamma |w - \hat{w}| \delta l \tag{36}$$

Those conditions are referred to a local frame of reference introduced in Fig.(5):

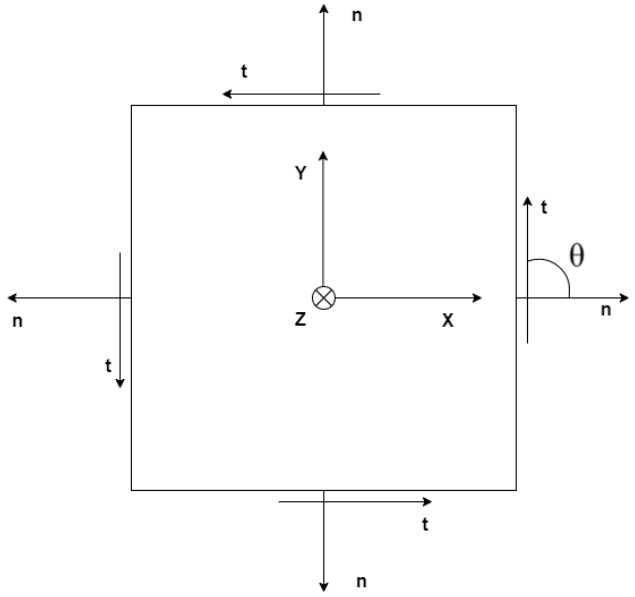


Figure 5: Local reference frame with respect to global coordinates. Normal and tangential directions indicated in every boundary of the panel.

where λ_n, λ_t and λ are the penalty factors in the normal, tangential and transverse directions, respectively. u_n, u_t and w refer to the displacement components in the same directions, and their $\hat{u}_n$, $\hat{u}_t$, $\hat{w}$ counterparts, the value of imposed displacements. This local reference frame is related to the Cartesian frame by a rotation that relies on angle θ indicated in the previous figure. The transformation is given in Eq.(37):

$$\begin{aligned} u_t &= sin(\theta) u_y + cos(\theta) u_x \\ u_n &= cos(\theta) u_y + sin(\theta) u_x \end{aligned} \tag{37}$$

When the condition is exactly satisfied, the term vanishes and the penalty has no further effect. The terms shown in Eq.(36) can be easily extended to the context of S-GUF in a general form. By using numerical integration over the kinematic boundaries, Eq.(36) becomes:

$$T = \lambda \sum_{j=1}^{Q_{bd}} \sum_{k=1}^{N_k} \sum_{p=1}^{N_p^k} [(u_n^{p,k,(j)} - \hat{u}_n^{p,k,(j)})^2 + (u_t^{p,k,(j)} - \hat{u}_t^{p,k,(j)})^2 + (w^{p,k,(j)} - \hat{w}^{p,k,(j)})^2] \frac{l_t}{Q_{bd}} \qquad (38)$$

where a common penalty factor λ has been considered for all displacement components, l_t is the total length of the edge and Q_{bd} the number of sample points (j) on which the expression is evaluated. The selection and influence on the solution of this penalty factor is covered in Section [5].

4 PINN-ELM framework

4.1 Physics Informed Neural Networks (PINN)

In this chapter, it is convenient to introduce previously some useful notation. Loss functions are referred with L, while the number of layers of a neural network are referred as Q and a specific layer index with its lowercase counterpart. Internal parameters of a network, are referred as w for weights and b for biases. Contrary to the previous chapter, where they were used to refer the transverse displacement component, and one of the panel characteristic dimensions. *g()* always refers to an activation function affecting the expression inside the parenthesis. Additionally, intermediate layers in a neural network between the input and output ones are referred as *hidden layers*.

This kind of network is a *feed-forward* network, meaning each layer creates data for the next layer when the specific nodes are activated by certain *activation functions*, that make the network non-linear to inputs. The basic operation of activation between layers is given in Eq.(39):

$$\boldsymbol{x}^q = g^q(\boldsymbol{w}^{q-1}\boldsymbol{x}^{q-1} + \boldsymbol{b}^{q-1}) \qquad (q = 1....Q) \tag{39}$$

$\boldsymbol{x}^q$ refers to the values of the nodes at the current layer, while $\boldsymbol{x}^0$ is the input data in the input layer, and $\boldsymbol{x}^Q$ to those in the output layer. The current layer is determined by index q and the number of layers used by the network is Q. $\boldsymbol{w}$ and $\boldsymbol{b}$ are internal parameters (matrices) to the network called weigths and biases, and *g()* refers to the specific activation function used (i.e. *hyperbolic tangent*). The expression indexes indicate that the activated values in a layer $\boldsymbol{x}^q$ are dictated by the assigned activation to that layer, and the product of weight and input data vectors plus the biases. This means that the information from a previous layer is combined with the values of all connections among nodes to determine the values in the next layer. This complex mapping is what makes the network to learn more complex features hidden in the data and better approximate its behaviour. The *training* process is indeed the adjustment of these internal parameters such that the loss function, L, is minimized.

In the context of supervised learning for a traditional neural network, the minimization of the cost function is done through a comparison between known (exact) measurements that constitute the *labeled data* and the estimations given by the network. The cost function is defined as:

$$L = \sum_{i=1}^{N} |u^{(i)} - u^{(i)^*}| \tag{40}$$

where u^* is a set of known solutions and u the predictions done by the network. N holds for the number of i points where this comparison is performed.

This means that the fitting of the network parameters during training is not tailored by any knowledge present of the physical system at hand. The basic idea for PINN is to modify the definition of the cost function to account for the environment: fulfillment of certain governing equations, presence of boundary conditions and presence of transient conditions (initial). Therefore a new form of the loss function is defined:

$$L = \sum_{i=1}^{N} |u^{(i)} - u^{(i)^*}| + \sum_{j=1}^{M} |r^{(j)} - 0| + \sum_{k=1}^{R} |B_{,x}^{(k)} - 0| + \sum_{h=1}^{T} |F_{,t}^{(h)} - 0| \tag{41}$$

for a classical differential boundary value problem, of the kind:

$$\begin{aligned} & r := u_{,t} + N_{,x}(u,x) = 0 \\ & B_{,x}(u,x) = 0 \qquad x \in \Gamma \\ & F_{,t}(u,x) = 0 \qquad t \in [t_0, t_f] \end{aligned} \tag{42}$$

where $N_{,x}$ is a functional of the differential governing equations of the problem, $B_{,x}$ and $F_{,t}$ are the boundary and initial conditions, respectively. All these extra terms are evaluated at different subsets of sample points, M,R, and T, in order to give the network with a distributed intuition on all the domain considered, both temporally and spatially.

The revolutionary feature introduced by PINN is precisely its awareness of the equations evaluated at the sampling points. Those equations incorporate partial derivatives of the output variables, that must be accounted for, thanks to the use of *Automatic Differentiation* and which allows to evaluate any-order derivatives at discrete points in space. Traditional ANN produces a complex fitting curve from raw input-output data, but does not understand if the predictions derived from this fitting are physically consistent, something that PINN is indeed penalizing in its loss function. PINN holds for a framework that dictates how the network operates by incorporating the physical governing relations of the problem in a hard way. This approach can be considered a class of Reinforcement Learning, provided it focuses on minimizing a cost. If during the training process, a governing equation is not satisfied, then the loss function is penalized and the adjustment of internal parameters will steer towards rectifying this tendency.

4.1.1 Automatic Differentiation

An ineludible part of any mathematical formulation in this field is the presence of derivative terms of different orders, that usually have to be evaluated over discrete datasets. A recently introduced way to achieve this is with the so-called *Automatic Differentiation* (AD) [3]. It stands for a family of techniques that avoids dealing with derivative expressions but instead uses numerical accumulation of terms during execution. Contrary to symbolic differentiation, it provides numerical values for the derivatives although it uses symbolic rules of differentiation at an inner level. It must not be confused with numerical differentiation, see for example finite difference methods. Whenever AD is to be used, is because an accurate numerical evaluation of the derivatives is indeed required. The effect known as *expression swell* that affects symbolic differentiation when extremely large expressions have to be evaluated in memory is not affecting AD, in the sense that only intermediate values and not the expressions themselves are stored. From an operational viewpoint, this fact has a significant influence and gives an insight on why this method is such an interesting choice in ML, where thousands of derivative evaluations should be performed on large datasets. This method makes use of the chain rule to combine the derivatives of constitutive items and get to the final evaluation. Two modes of operation are distingished: *forward* and *backward* modes.

A function f is specified to have n independent variables, namely x_i where $i = 1...n$, and m dependent variables that result from the evaluation of the function, referred as y_j for $j = 1...m$. The forward technique behaves better when the differentiation implies a reduced number of independent variables, and therefore few passes of the method are required. It implies that $n << m$. Considering the gradient of the target function that we want to differentiate:

$$\nabla f = (\frac{\partial y}{\partial x_1}, ..., \frac{\partial y}{\partial x_n}) \tag{43}$$

where a single output variable, y, is used in this example.

This whole procedure can be understood from the viewpoint of a computational graph in Fig.(6), where the intermediate operations are identified as nodes of a directed graph that combines their operations from given dimensional inputs. A primal trace is generated from the evaluation of the target function at a given point, and then used in subsequent derivative traces in a similar fashion.

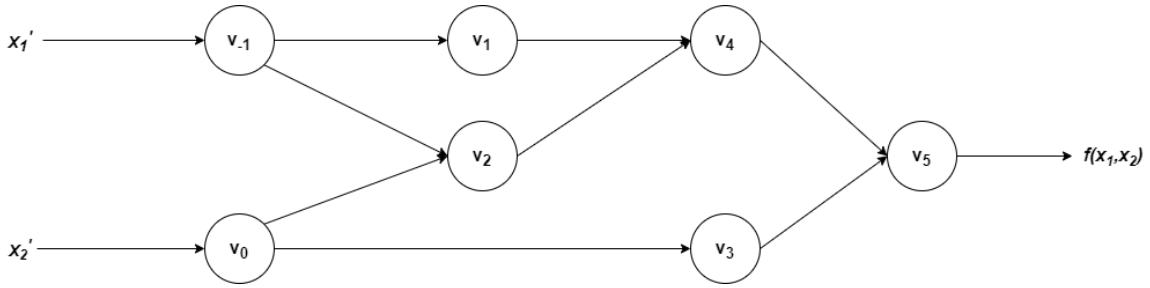


Figure 6: *Forward* AD directed graph - primal trace

The general additive procedure is illustrated in Table (1).The following table reports an example in which a generic function of two independent variables is given as $y = f(x_1, x_2) = ln(x_1) + x_1x_2 - sin(x_2)$, and differentiation is applied at a given point of interest, $(x_1^{'}, x_2^{'})$. The derivation is done with respect to x_1 by setting $\dot{x}_1 = 1$. The intermediate kernels are termed υ and take the values of the following column, the tangent operations on the right hand side of the table are preceded by the evaluations on the left.

Forward Primal Trace		**Forward Tangent Trace**	
Kernel	**Evaluation**	**Kernel**	**Evaluation**
υ_{-1}	x_1	$\dot{\upsilon}_{-1}$	$\dot{x}_1$
υ_0	x_2	$\dot{\upsilon}_0$	$\dot{x}_2$
υ_1	$ln(\upsilon_{-1})$	$\dot{\upsilon}_1$	$\dot{\upsilon}_{-1}/\upsilon_{-1}$
υ_2	$\upsilon_{-1} \times \upsilon_0$	$\dot{\upsilon}_2$	$\dot{\upsilon}_{-1} \times \upsilon_0 + \dot{\upsilon}_0 \times \upsilon_{-1}$
υ_3	$sin(\upsilon_0)$	$\dot{\upsilon}_3$	$\dot{\upsilon}_0 \times cos(\upsilon_0)$
υ_4	$\upsilon_1 + \upsilon_2$	$\dot{\upsilon}_4$	$\dot{\upsilon}_1 + \dot{\upsilon}_2$
υ_5	$\upsilon_4 - \upsilon_3$	$\dot{\upsilon}_5$	$\dot{\upsilon}_4 - \dot{\upsilon}_3$
y	υ_5	$\dot{y}$	$\dot{\upsilon}_5$

Table 1: Example of application of *Automatic Differentiation* in *forward mode* reported in [3].

Following this process, kernel $\dot{\upsilon}_5$ is obtained at the end of the tangent trace, which corresponds to the derivative of the function with respect to x_1.

Another possibility is to apply AD in a *backward mode*, which corresponds to a generalized backpropagation of derivatives from output to input. The primal trace maintains its forward behaviour and serves to populate the intermediate graph nodes (see Fig.(7)), while the node adjoints are employed to compute the derivative trace. Table (2) shows the same example function used previously for the forward mode:

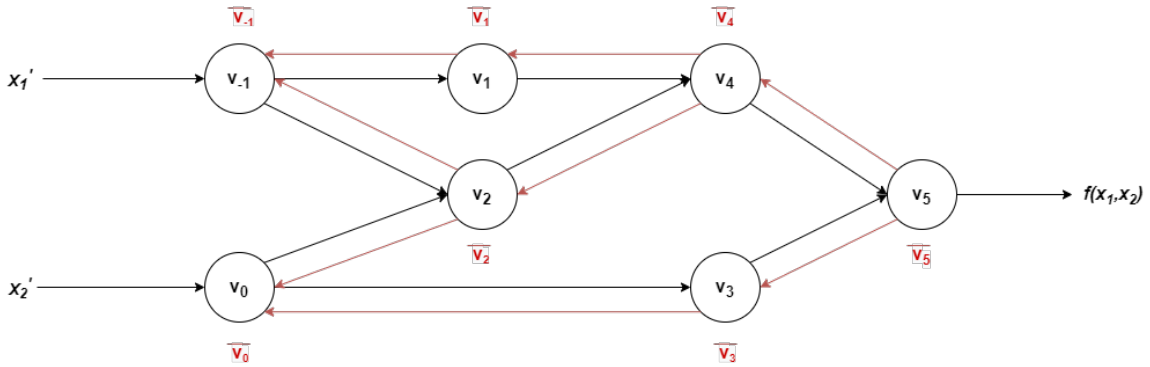


Figure 7: *Backward* AD directed graph - primal and adjoint traces

Forward Primal Trace		**Reverse Adjoint Trace**	
Kernel	**Evaluation**	**Kernel**	**Evaluation**
-	-	$\bar{x}_1$	$\bar{v}_{-1}$
-	-	$\bar{x}_2$	$\bar{v}_0$
-	-	$\bar{v}_{-1}$	$\bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$
v_{-1}	x_1	$\bar{v}_0$	$\bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$
v_0	x_2	$\bar{v}_{-1}$	$\bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$
v_1	$ln(v_{-1})$	$\bar{v}_0$	$\bar{v}_3 \frac{\partial v_3}{\partial v_0}$
v_2	$v_{-1} \times v_0$	$\bar{v}_2$	$\bar{v}_4 \frac{\partial v_4}{\partial v_2}$
v_3	$sin(v_0)$	$\bar{v}_1$	$\bar{v}_4 \frac{\partial v_4}{\partial v_1}$
v_4	$v_1 + v_2$	$\bar{v}_3$	$\bar{v}_5 \frac{\partial v_5}{\partial v_3}$
v_5	$v_4 - v_3$	$\bar{v}_4$	$\bar{v}_5 \frac{\partial v_5}{\partial v_4}$
y	v_5	$\bar{v}_5$	$\bar{y}$

Table 2: Example of application of *Automatic Differentiation* in *backward mode* reported in [3].

where the values of the adjoints $\bar{v}_{-1}$, $\bar{v}_0$ are obtained in the last step, and correspond to the derivatives of the function with respect to x_1 and x_2.

Along with the adjoints, the reverse operations require from the partial derivatives of the kernels of the left-hand side, thus, the initial evaluation of the primal trace complements the discovery of subsequent derivatives. An evident advantage over forward method is that it allows to get the derivatives in terms of all independent variables for a given point with just one pass of the algorithm. However, it becomes worse when the derivation is done with respect to few variables, because computing the adjoints is considerably more expensive than the forward tangent operations. Therefore, reverse mode is well suited for highly dimensional expressions, typically where $n >> m$.

In general, this method can be applied iteratively to unveil higher order derivatives. Provided the characteristics of this work, the low dimensionality (in terms of (x, y) coordinates only) of the 2D axiomatic approach makes it desirable to employ forward automatic differentiation for derivative evaluation, which is clearly more efficient from the computational side.

4.2 An overview of Extreme Learning Machines (ELM)

The introduction of the so-called *Extreme Learning Machine* algorithm tries to deal with the particular problem of slow learning speed for conventional training algorithms, and aims at providing good generalization performance with a much faster convergence. Traditional learning algorithms such as *gradient descent* suffer from some issues regarding training, that become particularly inconvenient when large datasets are handled and specially in the context of supervised learning. The aim of the learning algorithm is to find the global minimum of that function to complete training. It is performed by feeding the network with different batches or subsets of the training data from which it is able to learn more features and move in the context of this loss functional space towards minimization. However, in real application problems, the loss function to minimize is characterized by the presence of local minima and noisy target values. Therefore, the complexity of the minimization usually requires the selection of small values of the learning rate (or other related parameters) to facilitate the progression to the minimum in a smooth way, increasing considerably the computational time. Furthermore, other present problems like overshooting in the update when the value of the loss is refreshed, are provoked by a too large learning rate that makes the network lose the minimum point in the middle of two function evaluations and can even make it diverge. Or the trouble associated to not-normalized data, that require a normalization step before attempting to minimize the value of the loss, otherwise, the minimum point might be lost due to the bad numerical representation of the function space. In particular, by extrapolating these considerations into the field of PINN, it is reasonable to think of its minimization as a more intricate process. The loss function is penalized not only by the agreement with data pairs, but also rewarded by the fulfillment of the physical intuition of the problem. Therefore during the updating process inside training, the violation of a governing equation will lead to a large penalization of the loss function as mentioned in Section [4.1], and the network will update its parameters to fulfill that contraint. This constant changes in the convergence history of the loss function (it depends of the function to be approximated) will influence significantly the computational time, making it converge slower towards the minimum. All these issues, make it interesting to pay attention to other learning strategies that can leverage the complexity of selecting proper learning parameters and reduce computational time over classical gradient-based training.

The principle of ELM algorithm is settled on the idea of having a SLFN. A network of this kind with N hidden nodes and non-linear activations allows to learn N distinct observations without the need to adjust the parameters of the linking connection between input and hidden layer, but relying on a *random generation*. The reason behind this randomization is that it reduces the complexity of the problem to only determining the value of the weights connecting the output layer, turning the problem into a linear system that can be analytically determined. Mathematically, it is easy to understand that having a single hidden layer allows to reach the output with a single set of unknowns on the connecting output weights. Eq.(44) introduces the mathematical operations from input to output and compare them to available data:

$$\sum_{i=1}^{N} \beta_i g(w_i x_j + b_i) = t_j \qquad (j = 1....M) \tag{44}$$

The notation is slightly changed with respect to the previous chapter, due to the simplicity of SLFN networks, the indicial term q referred to the current layer is avoided. N is the number of hidden nodes and M the number of measurements available, the number of predictions. β_i states for the output weights, the unknowns of the system in the ELM framework, while w_i and b_i are the randomly generated weights and biases connecting the inputs with the hidden layer. Note this distinction between both sets of weights. t_j are the output values of the net-

work, which should be typically compared to actual datasets of previusly known exact results in order to monitor the learning process. g() specifies the use of a non-linear activation at the hidden layer, the activation of the output layer is always *linear* in the context of this training algorithm. Furthermore, the output layer is *unbiased*, meaning that the agreement with the predictions relies purely on the fitting of output weigths. Therefore, the left-hand side of the previous equation refers to the prediction done by the network using ELM training.

Based on this definition, the problem can be reformulated in terms of matrix notation such as:

$$\boldsymbol{H\beta} = \boldsymbol{T} \tag{45}$$

where H is called the *hidden layer output matrix* and T is the output matrix (vector), which includes all available output data for comparison.

$$\boldsymbol{H} = \begin{pmatrix} g(w_1 x_1 + b_1) & ... & g(w_N x_1 + b_N) \\ ... & ... & ... \\ g(w_1 x_M + b_1) & ... & g(w_N x_M + b_N) \end{pmatrix}$$

This training algorithm requires that the chosen activation function for the hidden layer is infinitely differentiable. If this condition is met, the training process reduces to minimizing the form of the linear problem as in Eq.(46):

$$min_\beta \|\boldsymbol{H}(w,b)\boldsymbol{\beta} - \boldsymbol{T}\| \tag{46}$$

A *minimum norm least-squares* solution can be found for the above system, but the treatment of the invertibility of matrix H changes for different situations. If $N = M$ then H is square and its invertibility only has to do with the presence of repeated samples, the problem can be approximated with MSE [5] zero. However, for the majority of real cases M is larger than the number of neurons N, thus having an equal number of neurons in the hidden layer is not realistic. In this case H is non-squared and the *Moore-Penrose pseudoinverse* (represented by †) has to be applied into solving the problem:

$$\begin{aligned} &\hat{\boldsymbol{\beta}} = \boldsymbol{H}^\dagger \boldsymbol{T} \\ &\|\boldsymbol{H}\hat{\boldsymbol{\beta}} - \boldsymbol{T}\| = \|\boldsymbol{H}\boldsymbol{H}^\dagger\boldsymbol{T} - \boldsymbol{T}\| = min_{\beta,\varepsilon}\|\boldsymbol{H\beta} - \boldsymbol{T}\| \end{aligned} \tag{47}$$

the solution will always maintain a threshold error ε that could be further reduced with a larger network, reaching a compromise in terms of complexity. $\hat{\beta}$ is one solution of least-squares that minimizes the error.

The extreme learning algorithm opens up an additional possibility that makes it an interesting choice used together with PINN. With no previously available data for training, it can get to a solution by simply fulfilling the physical intuition given by the loss function. Considering the framework of PINNs, a problem can be set up by discretizing the physical domain into a set of collocation points (where governing equations are applied) and boundary points (where boundary conditions hold), leading to a *boundary value collocation problem*. The solution to the problem corresponds to using a least-squares method which hidden layer output matrix is now embedded into the definition of the governing equations at every point.

[5] Mean Squared Error, this is the typical metric used to track performance on a learning algorithm, although not a unique choice.

4.3 Hyperparameters

Regarding neural networks application to structural analysis, the combined use of PINN with ELM constitutes the core of the implementation carried out in this work. To this matter, no available data of the solution field is used to minimize PINN's loss function, instead training is performed fulfilling the governing equations reported in Section [3]. In a neural network, a distinction can be made between *parameters* and *hyperparameters*. The *parameters* are coefficients of the model automatically optimized or selected during the learning process, they only require to be initialized. On the other hand, *hyperparameters* are manually selected by the operator, the model will not update them during the process. A clear example of this distinction is that of the network weights, which are themselves parameters once initialized, but the range of initialization is the hyperparameter selected by the user. This specific topic and others are covered along this chapter, where it is explained what every configuration means and how they are selected in this framework.

Weights and biases initialization

The process of randomly generating the weights and biases linking the input and hidden layer is performed with a given seed and a specific pseudorandom generator (by default *v5uniform*, available in MATLAB). The introduction of seeding in random processes ensures that although the generated sampling points are indeed random, the initialization point for the generator is known, what allows to reconstruct the generated sequence across different runs. The generator specifies the kind of distribution that the numbers fall into, this is the shape of the probability distribution of the generated samples. From this initial step, a matrix of random weights, $\boldsymbol{w}$, and random biases, $\boldsymbol{b}$, are available. These are then fed into the primal generation algorithm (see Section [4.1.1] about AD), that computes the values of the hidden nodes matrix, referred in this chapter as $\boldsymbol{X}_H$:

$$\boldsymbol{X}_H = g(\boldsymbol{w}\boldsymbol{P} + \boldsymbol{b}) \tag{48}$$

where $\boldsymbol{P}$ is a vector of point coordinates introduced in the input of the network. These coordinates are non-dimensionalized with respect to the Cartesian coordinate system introduced for SGUF, what allows to easily treat any panel geometry in a similar way. x coordinates non-dimensional counterpart is identified as χ and for y coordinates as η, both have been scaled down to range $[-1, 1]$ by dividing with the maximum coordinate value in both cases. It is this transformation what makes matrix $\boldsymbol{X}_H$ different from hidden layer output matrix $\mathbf{H}$ reported in Eq.(4.2), the conversion is done thanks to a redimensionalization strategy that uses half-characteristic dimensions of the panel:

$$\boldsymbol{H} = (\frac{\partial x}{\partial \chi})^0 (\frac{\partial y}{\partial \eta})^0 \boldsymbol{X}_H = (\frac{2}{a})^0 (\frac{2}{b})^0 \boldsymbol{X}_H \qquad (\boldsymbol{X}_H = \boldsymbol{X}_H(\chi, \eta)) \tag{49}$$

Subsequent derivations over the tangent traces are performed thanks to *Automatic Differentiation* for the network, allowing to compute the partial derivatives of $\boldsymbol{X}_H$. The redimensionalization to Cartesian coordinates is performed in the exact same way as in Eq.(49), the process is illustrated for first and second order derivatives:

$$
\begin{aligned}
\boldsymbol{H}_{,x} &= (\frac{\partial x}{\partial \chi})^1 (\frac{\partial y}{\partial \eta})^0 \boldsymbol{X}_{H,\chi} \\
\boldsymbol{H}_{,y} &= (\frac{\partial x}{\partial \chi})^0 (\frac{\partial y}{\partial \eta})^1 \boldsymbol{X}_{H,\eta} \\
\boldsymbol{H}_{,xx} &= (\frac{\partial x}{\partial \chi})^2 (\frac{\partial y}{\partial \eta})^0 \boldsymbol{X}_{H,\chi\chi} \\
\boldsymbol{H}_{,yy} &= (\frac{\partial x}{\partial \chi})^0 (\frac{\partial y}{\partial \eta})^2 \boldsymbol{X}_{H,\eta\eta} \\
\boldsymbol{H}_{,xy} &= (\frac{\partial x}{\partial \chi})^1 (\frac{\partial y}{\partial \eta})^1 \boldsymbol{X}_{H,\chi\eta}
\end{aligned} \tag{50}
$$

The previous matrices serve to represent the different degrees of freedom introduced by kinematic theories inside the network approximation. In this way, a kinematic variable and its subsequent space derivatives are given represented by these matrices, where each row correspond to a point in the domain and each column to the evaluation at every hidden neuron.

Adimensionalization and characteristic lengths

Regarding the introduction of matrix $\boldsymbol{H}$ and its space derivatives, the kinematic variables used in the model can be represented with a *network approximation* that relies on these matrices and the vector of output weights. Taking as an example an ED_{110} model, the displacement u_x is defined by two kinematic variables u_{x0} and u_{x1}, these are introduced into the network framework by identification with matrix $\boldsymbol{H}$. Their derivatives with respect to x and y are present in the strong-form governing equations and are analogously identified with $\boldsymbol{H}_{,x}$ and $\boldsymbol{H}_{,y}$. This example is given in Eq.(51):

$$
\begin{aligned}
\boldsymbol{u}_{x0} &= c_0 * \boldsymbol{H}\boldsymbol{\beta}_{x0} \\
\boldsymbol{u}_{x1} &= c_1 * \boldsymbol{H}\boldsymbol{\beta}_{x1}
\end{aligned} \tag{51}
$$

where $\boldsymbol{u}_{x0}$ and $\boldsymbol{u}_{x1}$ are vectors that contain the evaluation of those kinematic variables on every collocation point. $c_{\alpha_{ux}}$ are denoted as *characteristic lengths*, a pair of constants that are used to artificially inflate the value of that degree of freedom inside the network treatment. This is needed in very specific cases to treat problems where some terms are neglected by the network, although they keep a value of 1 by default in any other situation. $\boldsymbol{\beta}_{x0}$ and $\boldsymbol{\beta}_{x1}$ are subsets of the output weights vector $\boldsymbol{\beta}$, that correspond to the connection between the hidden nodes and those specific output nodes.

Network connections configuration

The versatility of PINNs permits them to tackle with different network architectures. *Dense* networks, understood as fully mutual connections between all nodes in two adjacent layers is the preferred option when modelling physical systems. Each neuron is influenced by all inputs from the previous layers, what translated downstream indicates a complete interconnection between all inputs and outputs. *Sparse* networks on the contrary have little application (it is a recently new field) and will not be considered in the context of PINN. An example of a dense network for structural analysis is reported in Fig.(8):

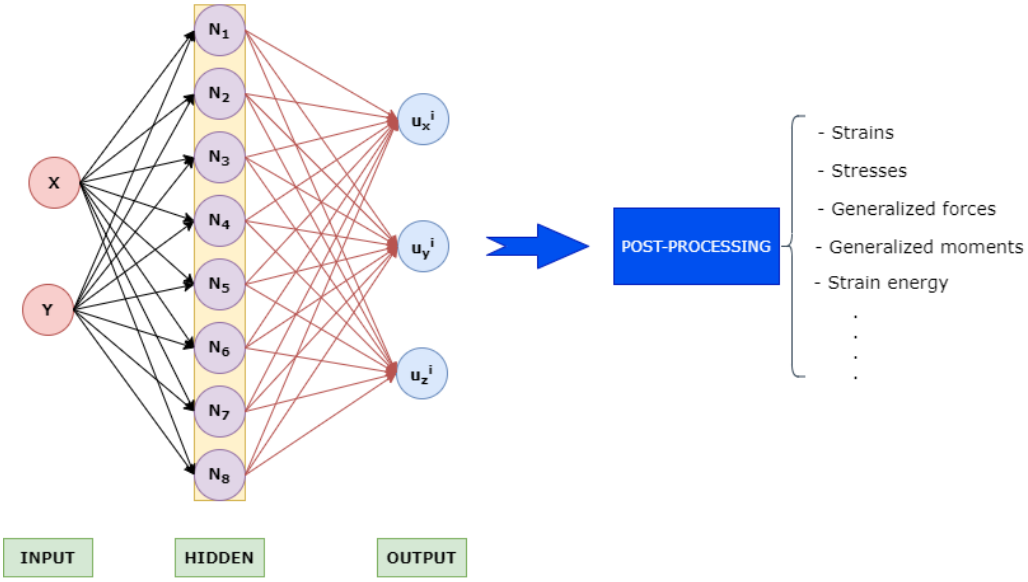


Figure 8: SLFN architecture in a displacement-based PINN network approach.

where eight nodes form the hidden layer. The input is given by the physical coordinates of a point, and the outputs are the displacement components at that same point.

Moreover, it corresponds to the architecture of a dense SLFN (concept firstly introduced in Section[1]), where only a single hidden layer is present. Similarly, a SLFN can be turned into a multi-network, what allows to have a better individual control on the *computational effort* allocated to each degree of freedom. Multi-networks allow for a non-equal partition of neurons, more effort can be allocated to compute specific outputs, while keeping training complexity to be the same. An example regarding the previous architecture is reported in Fig.(9), where the network has been splitted on the different displacement components:

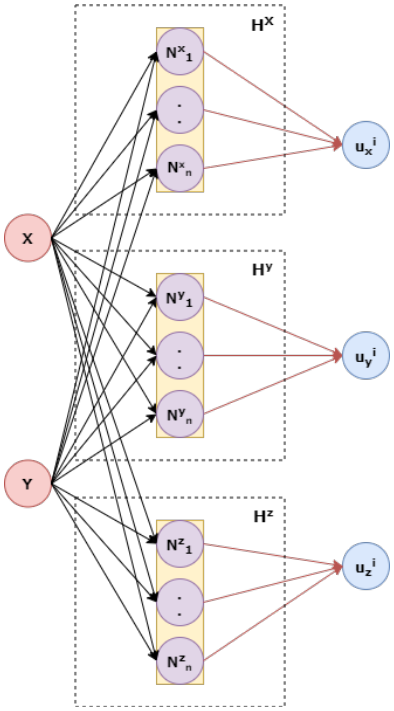


Figure 9: Alternative multiple network architecture for a displacement-based PINN network approach.

H^x, H^y and H^z are referred to the groups of neurons allocated to each output node. This strategy has been selected as a preferred network architecture in both methods, as it allows for an individualized selection of the hyperparameters. All results shown in Section [5] consider this kind of architecture, although the solution itself is not influenced by it. It has been considered a good addition regarding the development of further works based on this.

Activation functions

The next consideration comes to the selection of the activation functions that originate those $\boldsymbol{H}$ matrices employed for the network approximation. The value of a hidden layer node in a network is determined by a trivial mathematical operation involving the information from the previous layer, and the weights and biases that connect to that node. The performance of the network to represent and learn the information of the model is directly influenced by the function chosen, which plays an active role on the prediction accuracy of the trained network. There are many possibilities for the activation function to fit better or worse depending on the problem to be solved. In the context of ELM, the connection between the input and hidden layer is ruled by a non-linear activation, such that it allows the model to create complex mappings between network inputs and outputs. One of the reasons to select a non-linear activation over a linear one for this intermediate layer is motivated by the fact that governing equations usually comprehend high order derivatives that must be evaluated using automatic differentiation. To express the problem using the network approximation, space derivatives must be handled, and to this purpose the hidden layer output matrix derivatives are obtained (see Eq.(50)). Because this matrix involves the values of the activated nodes, the activation functions are also derived in the chain rule, therefore a function to be used must be differentiable up to the required extent by the problem, while a linear connection is only differentiable once to a constant value. The output layer on the other hand is restricted to the use of a linear activation by ELM, what allows to use solution methods for linear systems.

Several non-linear activations are introduced as alternatives to be used within ELM in Fig.(10).

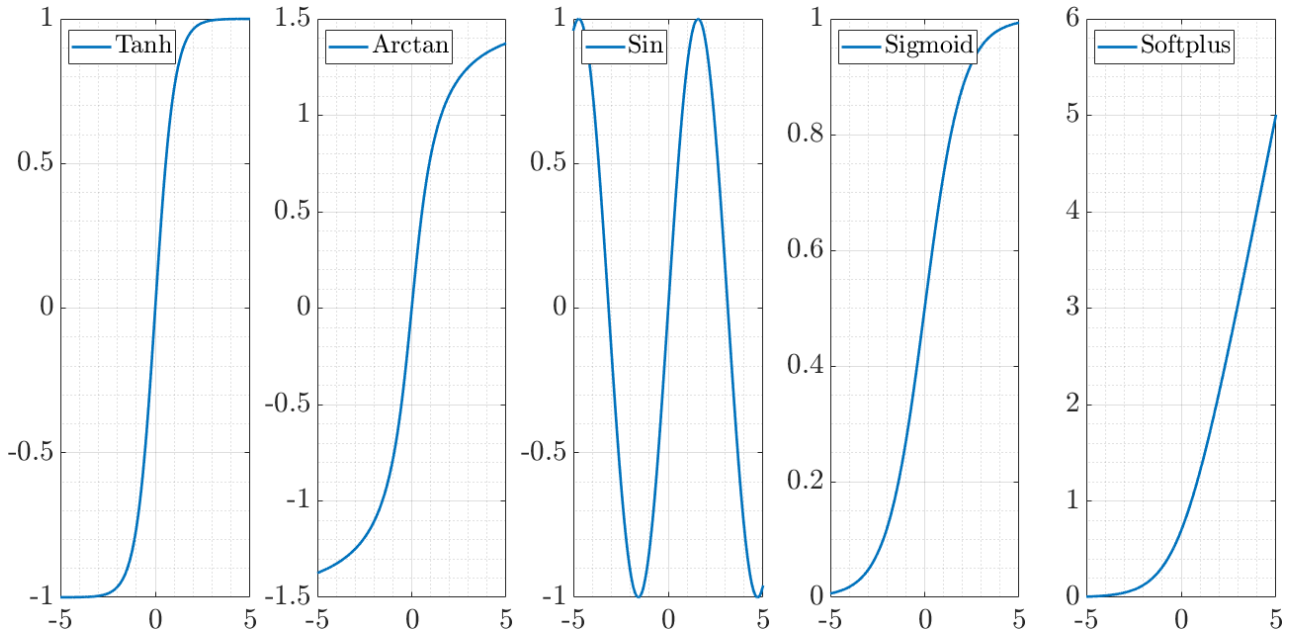


Figure 10: Non-linear activation functions: *hyperbolic tangent*, *arctangent*, *sine*, *sigmoid* and *softplus*.

Out of them, *tanh* and *sigmoid* are typically referred as archetypical functions in literature,

used in most applications. They are specially considered during the numerical assestment of Sections [5.4]-[5.5] as representatives of two kinds of behaviours observed during training. The other functions are also available in the program to be used.

Range of random initialization

The set of equations to be solved has a marked influence on the optimum selection of hyperparameters. A special attention is given considering the range for random generation of weights and biases within the network. These values will affect specifically the neural network by provoking a number of activated nodes to approach the saturation limits given by their activation function. Modifying these limits is meant to directly affect the capacity of the neural network to represent the physical problem at hand, allowing to use activated values in a wider range than those given by the empirical ($[-1, 1]$) range of weights and biases. These hyperparameters must be treated carefully to not hinder the heterogeneity of the connections with the input layer, which is the basis of the network representation. If the range considered is too broad, a large part of the generated values would create large inputs to the activation function when combined with input data. Many of them would approximate saturation limits and the numerical representation of the nodes would be highly biased to extremal values, losing the capacity to represent more complicated effects in the solution field.

Number of neurons

A common topic regarding neural networks has to do with the ways to select an appropiate number of neurons for the hidden layers. In a SLFN, enough neurons are required to properly represent the complexity of the solution, while an excessive amount can have the opposite effect incurring in a sort of *overfitting* due to a specially bad conditioning of the stiffness matrix. No threshold exists on wheter a problem is poorly represented, a convergence study is typically used to find the best configuration on every case. In the context of ELM, it was mentioned in Section [4.2] how a network can learn as many features as the number of neurons present in the hidden layer. This premise is generally valid, but becomes unclear when the target is to find the displacement components solution field. In the context of the strong-form formulation, the governing equations are evaluated at N^2 points, as a rule-of-thumb, a number of neurons equal or superior is taken for reference and expected to give a good representation of the behaviour of the solution. It implies the neurons are in the order of hundreds for most of the cases considered. The exact optimal point is also expected to vary between cases and depending on factors such as the boundary conditions, material or geometry of the panel. Shifting the problem to that of free vibrations is also expected to require a change in the number of neurons, in order to capture the different natural frequencies and their associated transverse deflection shapes. All of these considerations are covered extensively in Section [5].

Collocation points distribution

One extra element that must be introduced in the context of hyperparameters has a great influence on the solution of both methodologies, it is the shape (and denseness) of the collocation grid. A major concern has to do with the proper sampling of both the collocation (*interior*) region and the boundary (*edges*). An insuficient sampling of the domain borders corresponds to a scenario where the boundary conditions are not totally satisfied on the boundary region of interest. This can indeed lead to a wrong estimate of the displacement field, provided that a part of this subdomain, referred as Γ, would remain unconstrained. On the other hand, an excessive spacing of the grid points in the collocation region may lead to inaccurate interpolations, losing some features as well. A large sampling can however hinder the applicability

of this approach, leading to unreasonably long computational times, provided the equilibrium conditions are imposed on each collocation point. When this sampling process is translated to the frame of network approximation, the problem rapidly grows in size as each equation defined at every point has its own contribution on the weighting of all hidden layer nodes.

Four different discretization types have been considered for use in this project:

- **Uniform:** classical distribution, based on the uniform sampling of the domain at regular intervals. This is a preferred option to ensure a correct evaluation of surface/line integrals using numerical integration in the context of the energy method.
- **CGL:** it is a distribution on *Chebyshev-Gauss-Lobatto* nodes. These nodes are assembled according to trigonometric relations:

$$\begin{aligned} \chi_i &= [1 - \cos\frac{(i-1)\Pi}{N_{pt}^{(1)} - 1}] - 1 \qquad (i = 1...N_{pt}^{(1)}) \\ \eta_j &= [1 - \cos\frac{(j-1)\Pi}{N_{pt}^{(2)} - 1}] - 1 \qquad (j = 1...N_{pt}^{(2)}) \end{aligned} \tag{52}$$

  where $N_{pt}^{(1)}$ and $N_{pt}^{(2)}$ are the number of grid points used in χ and η directions (non-dimensional). This definition is based on the roots of *Chebyshev polynomials* of the first kind, a mathematical sequence of polynomials related to sine and cosine functions, and they are a usual selection for polynomial interpolation.

- **SCN:** states for *Scaled Chebyshev Nodes*. It is a slight modification of CGL that aims to improve the value of the Lebesgue constant [6] associated to the distribution. The grid generation equations are described in a similar manner:

$$\begin{aligned} \chi_i &= [1 - \frac{\cos\frac{(2i-1)\Pi}{2N_{pt}^{(1)}}}{\cos\frac{\Pi}{2N_{pt}^{(1)}}}] - 1 \qquad (i = 1...N_{pt}^{(1)}) \\ \eta_j &= [1 - \frac{\cos\frac{(2j-1)\Pi}{2N_{pt}^{(2)}}}{\cos\frac{\Pi}{2N_{pt}^{(2)}}}] - 1 \qquad (j = 1...N_{pt}^{(2)}) \end{aligned} \tag{53}$$

- **RNG:** states for a random normal distribution. By making use of a random number generator, a set of $N_{pt}^{(1)}$ and $N_{pt}^{(2)}$ points are generated in each direction. An associated inconvenient to this alternative is on its non-repeatibility of the sampling between different runs. Moreover, in some cases not enough points are available to map the boundaries, and it experiences a remarkable drop in performance.

A representation of the one-dimensional distribution of these sampling techniques is shown in Fig.(11) for 20 points:

[6] It is a constant which value is scaled within $[0, 1]$ and gives an idea of how good the interpolant of a function is compared to the best polynomial approximation of such function.

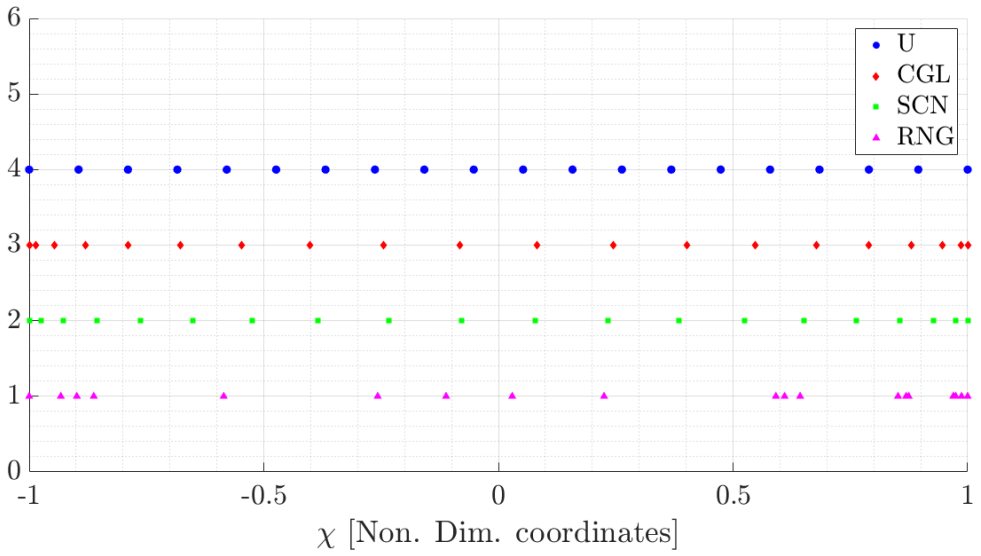


Figure 11: Different collocation distributions in a single direction for $N = 20$ sample points.

Different test results for these techniques are shown in Section [5] regarding the first collocation method. The most stable configuration is seeked in order to be considered a default option. The number of collocation points is supposed to have a great influence over the results, and it will be tested for different configurations of internal network parameters.

4.4 Training process

The training of the network with ELM in the context of both methods is made up by several contributions. The network approximation is substituted into each formulation, leading to different matricial components that are later assembled together. The product of the assembly process is a stiffness matrix then used by the LSQ solver to find the vector of output weigths. Finally, the displacement solution field is reconstructed and other quantities are derived from physical relations. Two problems are treated: a *boundary value* problem, intended to cover the linear static analysis of the panel, and an *eigenvalue* problem, which is set up starting from the same formulations already developed, aimed to find the natural frequencies of vibration.

4.4.1 Boundary value problem

In order to setup the training process, a system of linear algebraic equations is required. Once the network approximation is available for every kinematic variable, the governing equations can be assembled in a matricial way where the different contributions, constitutive terms ($\tilde{C}$) and integral kernels (Z) are constants multiplying these matrices. An overall *stiffness matrix* ($\boldsymbol{A}$) is obtained by combining all the different contributions. This assembly process is different depending on the formulation that is used, Fig.(12) shows the assembled matrix in the context of the strong formulation collocation method with regard to the example introduced in Eq.(51):

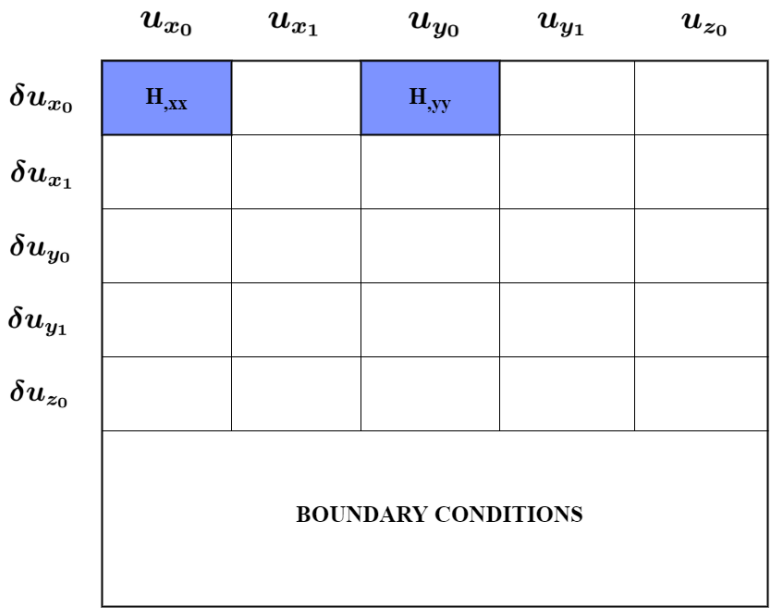


Figure 12: Example of assembly of the system of equations for the strong-form formulation at network level implementation. Case of plate subjected to FSDT theory and single ply, superscripts are not considered in this case.

The variational components originary of each equation are spanned in the row direction, while the last rows belong to the equations that introduce the boundary conditions, both of natural and essential kind. In this example, the first equation has two blocks blue coloured, corresponding to the network approximation of second order derivatives of two of the kinematic variables involved. It can be seen how the information of different variables is given in different column blocks when it comes to assembly inside a same equation. In a more complex case of multilayered contributions, rows and columns can be removed to enforce the interlayer compatibility at network level, which is the actual application of the strong formulation method compatibility conditions. In a similar way, a vector of external loads ($\boldsymbol{e}$) is obtained by combining in a vector the value of the load on every collocation point of the domain and particularizing this vector to the corresponding equation (i.e. uniform pressure corresponds to first kinematic variable equation for the transverse displacement).

In the context of the energy method, a square matrix is obtained from substituting the matrices of the network approximation into the strain energy (U) statement. A sketch of this process is shown in Fig.(13) for the same example of the previous case:

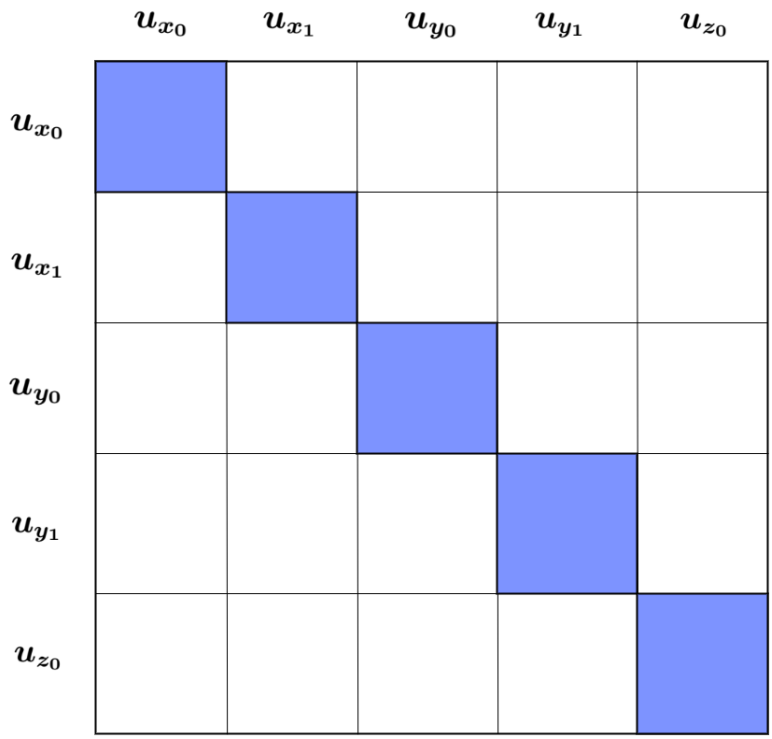


Figure 13: Example of assembly of the system of equations for the weak-form formulation at network level implementation. Case of plate subjected to FSDT theory and single ply, superscripts are not considered in this case.

The diagonal terms are highlighted to remark the differences with the first method, those terms correspond to the pure membrane and bending components, while off-diagonal terms are referred to coupling contributions. Therefore, the network approximation must deal with the product of two hidden layer output matrices (or its derivatives) for each contribution, and the assembled energy matrix is expected to be (quasi) completely full, as opposed to the case of the strong approach where the matrix is typically sparse. Moreover, the same approximation is done over the external contribution term, V, and the penalty contribution, T at network level. The penalized terms from the latter go in correspondence of the diagonal terms, due to the quadratic implementation shown in Eq.(38).

Once the stiffness matrix and the vector of external loads are known, the strategy to train both methods is the same, it passes through the solution of the least-squares problem. The training process in the context of ELM is performed solving a system of algebraic equations in the form of:

$$\begin{aligned} \boldsymbol{A\beta} &= \boldsymbol{e} \\ min_{\beta} &\|\boldsymbol{A\beta} - \boldsymbol{e}\| \end{aligned} \tag{54}$$

where $\boldsymbol{A}$ holds for the assembled stiffness matrix, $\boldsymbol{e}$ is a vector containing the value of the external stimuli at each collocation point and $\boldsymbol{\beta}$ is the vector of *output weights*, the unknowns to be found. For the strong formulation, $\boldsymbol{A}$ is meant to be the assembly of all governing equations, while in the energy method, it refers to $\boldsymbol{A} = \boldsymbol{U} + \boldsymbol{T}$, where $\boldsymbol{U}$ and $\boldsymbol{T}$ are the network approximations (matrices) of the strain energy and penalty contributions. In the latter, vector $\boldsymbol{e}$ is also meant to identify the network approximation of the external term, $\boldsymbol{V}$.

In the context of the strong formulation, the problem is either *overdetermined* or *underdetermined* and it has no unique solution. Both scenarios are possible and it will depend directly on

the size of the collocation grid and the hidden layer, and indirectly on the orders of expansion of the model chosen to simulate. The process is narrowed down to an individual choice, by imposing an extra condition on the selection of a solution with the minimum norm:

$$min(\|\hat{\boldsymbol{\beta}}\|) \tag{55}$$

where $\hat{\boldsymbol{\beta}}$ is that set of weights fulfilling this condition.

Before solving the system, the matrices from the network approximation must be assembled, after which they undergo a *normalization* process. This is a common technique used for neural networks, which tries to improve the way in which training samples (matrix components in this case) are numerically treated in the objective minimization process. The values in the matrix present a large numerical disparity, mainly due to the inclusion of the constitutive relation terms, which are originally expressed in (large) physical units. Several well-known strategies are available in the literature such as: *standard* scaling, *min-max* scaling and *max-abs* scaling. Due to its simplicity of implementation, a *max-abs* is used as a default in all methods. This strategy divides all components of a row in the matrix by the absolute value of the maximum of that row.

Furthermore, it is reasonable to think that in a collocation problem, the assembly matrix will be naturally ill-conditioned. Ill-conditioning occurs whenever a matrix posseses very similar row values, which can be a result of the evaluation of the governing equations at two points that are very close to each other. This can be overcome by different invertibility techniques such as the pseudoinverse, although the major concept of ill-conditioning means that small changes in the inputs provoke a large variation in the answer (solution of the least-squares problem) what translates into a major difficulty to get to the expected solution. In this sense, scaling and normalization play a critical role, and make the problem to be less susceptible to ill-conditioning. The same is applicable for the energy method, scaling is required to go into the solver and the *loss* value is maintained as an indication of the convergence of the solution.

After training, the displacement fields are recovered by inverting the network approximation with the known values of the output weights. This is equivalent to evaluating the expressions of the kinematic variables in Eq.(51). Once the displacements are known, strains and stresses can be recovered from the application of the strain-displacement and constitutive relations.

In order to test how well a solution found fits within the original problem, a *residual* metric has been defined, that evaluates the deviation of a resulting output weight to fit the linear system equality:

$$\boldsymbol{R_{es}} = abs(\boldsymbol{A\beta} - \boldsymbol{b}) \tag{56}$$

$\boldsymbol{R_{es}}$ is a column vector with each row corresponding to the residual of a given equation in the system. Analogously, a *loss* metric is defined for the overall case as the squared sum of square-residuals normalized over the number of rows (contributions) of the linear system, and known as *Root-Mean Squared Error* (RMSE):

$$RMSE = \sqrt{\sum R_{es}^2}/N_{rows} \tag{57}$$

Additionally, the condition number of the stiffness matrix is reported as an indication of the accuracy of the results from matrix inversion and solution to the linear equation. The two-norm condition number is employed, introduced in Eq.(58):

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\| \tag{58}$$

A large value of this matrix indicates it is poorly conditioned (close to singularity), what translates into a major difficulty for the solver to find the seeked solution among other possibilities. The lower the condition number, the better conditioned the matrix of interest is.

4.4.2 Eigenvalue problem

Finding the natural frequencies and the corresponding modes of a structure is one of the best ways to analyze the capability of a component to be used for a certain application. Typically, it is of interest to design a structure which natural frequencies are well separated from the excitation frequencies induced by external loads at its nominal operation point, usually with an embedded safety margin. The solution to finding these frequencies and their associated modes is carried by means of an *eigenanalysis*, that relies on the very same foundation of the PINN-ELM linear static framework. The extent of the current application covers so far the solution of a real *eigenproblem*, where the stiffness matrix of the problem is real and symmetric. A generic form of the eigenproblem is given by Eq.(59):

$$[\boldsymbol{K} - \omega^2 \boldsymbol{M}]\boldsymbol{u} = \boldsymbol{0} \tag{59}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{M}$ is the mass matrix and ω is the value of the natural frequency, target of the problem. $\boldsymbol{u}$ is a vector of amplitudes/unknowns, that in this case can be identified with the displacement components.

Although the problems at hand covered along this thesis are referred to linear elasticity problems and the stiffness matrix will be always real, other situations can occur, for which specific modeling tools are available. For instance, for viscoelastic materials, the imaginary part of the matrix is meant to introduce damping, $\boldsymbol{K} = \boldsymbol{K}_R + j\boldsymbol{K}_I$, which indeed requires a different approach to the problem, not reported in this dissertation.

In this framework, the form of the problem is similar for both approaches. Within the strong formulation method, the final shape of the LSQ problem is of the form:

$$[\boldsymbol{A} - \boldsymbol{\lambda M}]\boldsymbol{\beta} = \boldsymbol{0} \tag{60}$$

where $\boldsymbol{\beta}$ holds for the vector of unknown output weights of the network, and $\boldsymbol{\lambda}$ refers in this context to a matrix containing the eigenvalues. Matrix $\boldsymbol{A}$ is not symmetric neither square, provided its number of rows is determined by the number of collocation evaluation points and equations, and columns are dictated by the number of neurons present in the hidden layer of the network. For this reason, a preconditioning matrix, $\boldsymbol{P}$ is introduced to make the problem suitable for an eigensolver. Matrix $\boldsymbol{M}$ is the mass matrix of the plate, that includes its inertial properties. It is defined as $\boldsymbol{M} = [\boldsymbol{M}_{\alpha_{u_x}}, \boldsymbol{M}_{\alpha_{u_y}}, \boldsymbol{M}_{\alpha_{u_z}}]^T$, a set of submatrices that include all multilayer contributions referred to each principal direction. Regarding the strong-form derivation from the original statement of the PVW, it is made of standalone mass functions regarding every kinematic variable and assembled in the same way as the stiffness matrix:

$$\begin{cases} \boldsymbol{M}_{\alpha_{u_x}} = \rho^{p,k} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux}} \\ \boldsymbol{M}_{\alpha_{u_y}} = \rho^{p,k} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy}} \\ \boldsymbol{M}_{\alpha_{u_z}} = \rho^{p,k} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz}} \end{cases} \tag{61}$$

The preconditioner premultiplies both the stiffness and mass matrices, improving the numerical performance to find the natural frequencies and modes. Two different preconditioners are available in the strong approach, the stiffness matrix transpose, $\boldsymbol{P} = \boldsymbol{A}'$, and the stiffness matrix

pseudoinverse, $\boldsymbol{P} = \boldsymbol{A}^{\dagger}$. Their performance is compared and reported in Section [5] for specific cases of application.

Regarding the weak-form formulation, the kinetic energy has to be introduced in order to account for the inertial properties of the panel. The assembly of the transient terms coming from PVW statement through numerical integration and into the network approximation ends up forming the energy counterpart of the *mass matrix*, M. The statement of this matrix is very similar to that in the first collocation method, with a slight change in notation regarding the off-diagonal contributions:

$$\begin{cases} \boldsymbol{M}_{\alpha_{u_x}\beta_{u_x}} = u^{p,k}_{x\alpha_{ux}} \rho^{p,k} Z^{p\alpha_{ux}\beta_{ux}}_{u_x u_x} u^{p,k}_{x\beta_{ux}} \\ \boldsymbol{M}_{\alpha_{u_y}\beta_{u_y}} = u^{p,k}_{y\alpha_{uy}} \rho^{p,k} Z^{p\alpha_{uy}\beta_{uy}}_{u_y u_y} u^{p,k}_{y\beta_{uy}} \\ \boldsymbol{M}_{\alpha_{u_z}\beta_{u_z}} = u^{p,k}_{z\alpha_{uz}} \rho^{p,k} Z^{p\alpha_{uz}\beta_{uz}}_{u_z u_z} u^{p,k}_{z\beta_{uz}} \end{cases} \tag{62}$$

This definition can be used indistinctly in the context of plates and shells within the energy method.

The contributions are then recombined to form an eigenproblem where the stiffness matrix is made up by the internal strain energy and any penalty term corresponding to the applied essential conditions, $\boldsymbol{A} = \boldsymbol{U} + \boldsymbol{T}$:

$$[(\boldsymbol{U} + \boldsymbol{T}) - \boldsymbol{\lambda M}]\boldsymbol{\beta} = \boldsymbol{0} \tag{63}$$

The eigenproblem can then be naturally solved, but better results are found by employing a preconditioner strategy. Also in this case, the transverse and pseudoinverse preconditioners have been tested and a wide comparison is stablished in Section [5].

5 Results

Major findings are shown in this chapter for both methodologies, implemented in different MATLAB codes. External results have been obtained for validation from the available implementation in MATLAB of a Ritz-SGUF approach, and a finite element commercial software called *ANSYS Mechanical*. Composites stackup sequences and modelling of plies are available in *ANSYS ACP* module, which is used whenever a validation of this kind is required.

The presentation of the results is organized as follows: a validation of both approaches, both in the context of linear static analysis and free vibrations is considered with several reference configurations shown in Section [5.1]. Certain numerical issues discovered during implementation are discussed in detail along with the situations when they take place and possible solutions are discussed in some cases. They are part of a broader numerical study with the target of identifying the influence that network hyperparameters have on the solution. Furthermore, a broad sensitivity analysis is included, which treats the way that several parameters interact together in order to better undertand their implications. Finally, a computational assestment is introduced for different setups regarding the available hardware.

5.1 Model validation: introduction

Different materials are employed to exemplify the capabilities of the methods, a compendium of them is reported in Tables [3]-[4] with their mechanical properties.

Material	E_1 [GPa]	E_2 [GPa]	E_3 [GPa]	ν_{12}	ν_{23}	ν_{13}
Al7050-T7451 (1) [65]	71.7	71.7	71.7	0.33	0.33	0.33
Invented isotropic (2)	100	100	100	0.30	0.30	0.30
Common alluminum (3)	73	73	73	0.33	0.33	0.33
Graphite epoxy (4) [2]	137.90	8.96	8.96	0.30	0.49	0.30
T-graphite (5) [2]	185	10.5	10.5	0.28	0.28	0.28
Pagano (6) [66]	172.37	6.89	6.89	0.25	0.25	0.25
Rao-Desai (7)	131	10.34	10.34	0.22	0.22	0.49
Ibrahim (8)	206.84	5.17	5.17	0.25	0.25	0.25
Meyer-Piening (face) (9)	70	71	69	0.30	0.30	0.30
Meyer-Piening (core) (10)	0.003	0.003	0.0028	0.25	0.25	0.25

Table 3: Mechanical properties of materials of interest. *First part.*

Material	G_{12} [GPa]	G_{23} [GPa]	G_{13} [GPa]	Density [kg/m^3]
Al7050-T7451 (1) [65]	26.9	26.9	26.9	2830
Invented isotropic (2)	5	5	5	1500
Common alluminum (3)	26.78	26.78	26.78	2860
Graphite epoxy (4) [2]	7.10	6.21	7.10	1450
T-graphite (5) [2]	7.3	7.3	7.3	1600
Pagano (6) [66]	3.45	1.38	3.45	1500
Rao-Desai (7)	6.89	6.20	6.89	1627
Ibrahim (8)	5.17	5.17	5.17	1603
Meyer-Piening (face) (9)	26	26	26	2700
Meyer-Piening (core) (10)	0.001	0.001	0.001	20

Table 4: Mechanical properties of materials of interest. *Second part.*

Notation number (1)-(10) is used to refer to tabulated materials in a compact way.

An illustration of the way boundary conditions are imposed in a panel is given in Fig.(14):

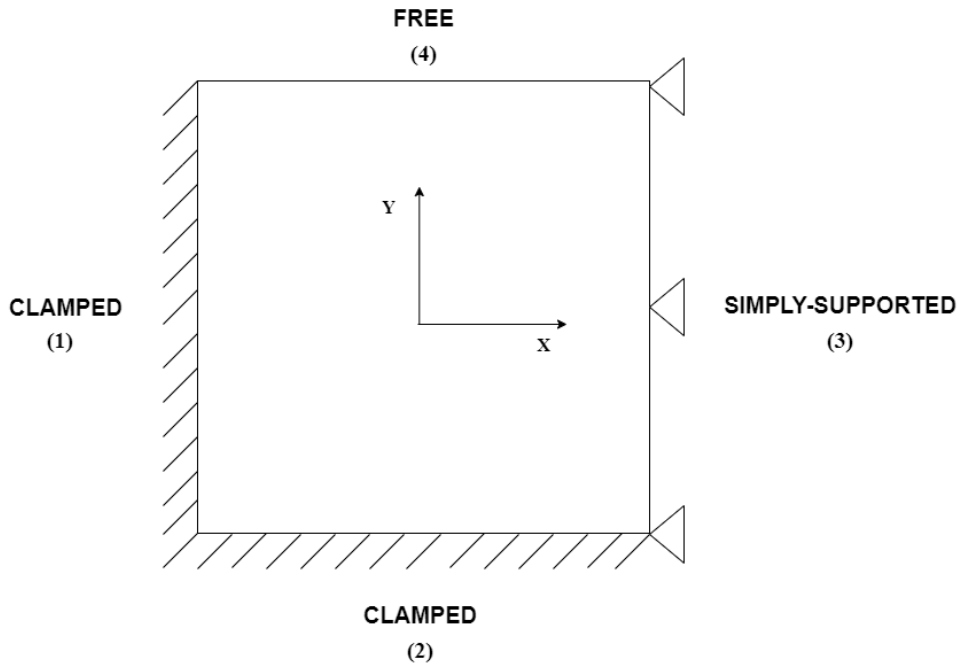


Figure 14: Square plate - Example of application of boundary conditions

Note the frame of reference is rotated counter-clockwise with respect to the frame introduced in Section [3], change motivated by a more intuitive way to handle boundary conditions, but that does not modify the formulation. The numbering in the edges is taken as a reference for the order in which a sequence of boundary conditions is given. In this case, the conditions are announced as CCSF, convention that always starts from the edge on the left.

Regarding variable-kinematic theories, boundary conditions are referred using the same nomenclature here: *clamped* (C), *simply-supported*(S) and *free* (F). More complex combinations of conditions are possible, but the analysis has been restricted to these three conditions for simplicity, that are representative of the great majority of situations that can be found. Kinematic theories are referred in the same way reported in Section [3], where equivalent single layer theories used the notation $ED_{N_{u_x}(Z)N_{u_y}(Z)N_{u_z}(Z)}$, and layerwise theories $LD_{N_{u_x}(E)N_{u_y}(E)N_{u_z}(E)}$. Subscripts are referred to the order of expansion of the theory in each axis, and (Z), (E) regard the introduction of zig-zag or partially layerwise models, if present.

A series of reference cases are considered to guide the discussion, both geometrical, material and neural network parameters are reported. Ritz validation cases are meant to use the same setup configuration regarding materials, geometry or kinematic theory, only the specific number of trial functions in each axis is reported. Validation of cases with FEM will also include the type of mesh and number of elements to use. In the context of the *strong-form formulation method*, the following configurations are used:

- **Configuration 1:** a thin alluminum (1) isotropic plate of dimensions $100 \times 100 \times 1mm$ is placed under a uniform load of $F = 1000N$ in magnitude directed in the negative Z-axis direction. A CGL grid of 20×20 collocation points, 500 neurons with *tanh* activation function are considered in the network setup. Kinematic theory ED_{110} is used as reference, if not specified otherwise. Ritz validation: 50×50 trial functions.

- **Configuration 2:** a three-ply multilayered plate of dimensions $100 \times 100 \times 1mm$ is placed under a uniform load of $F = 1000N$ in magnitude directed in the negative Z-axis direction. Kinematic theory ED_{110} is used for simulation. A CGL grid of 20×20 collocation points, 500 neurons with *tanh* activation function is considered in the network setup. Material combination and layup sequence are left as free parameters in the analysis. FEM validation: 30000 elements in a structured mesh, same geometry and lamination properties.

- **Configuration 3:** a two-ply composite (11) multilayered plate of dimensions $100 \times 100 \times 1mm$ and layup sequence $[0, 90]^\circ$ with equal-thickness plies is considered. A CGL grid of 20×20 collocation points, 500 neurons with *tanh* activation function is considered in the network setup. Kinematic theory ED_{110}. FEM validation: 30000 elements in a structured mesh, same geometry and lamination properties.

In the context of the *weak-form formulation method*, different cases are used to extend the discussion:

- **Configuration 4:** a thin alluminum (1) plate of dimensions $100 \times 100 \times 1mm$ is simulated under $F = 1000N$ uniform load directed in the negative Z-axis direction. ED_{110} kinematic theory is used, and a uniform grid of 100×100 sampling points is employed for numerical integration. Penalty factor $\lambda = 10^{10}$, 200 neurons and *tanh* activation. Ritz validation: 50×50 trial functions. This setup is the same as *Configuration 1*, with network parameters specific to the second method.

- **Configuration 5:** a three-ply multilayered plate of dimensions $100 \times 100 \times 1mm$ is placed under a uniform load of $F = 1000N$ in magnitude directed in the negative Z-axis direction. A uniform grid of 70×70 sampling points is employed for numerical integration, 200 neurons with *tanh* activation. Two different cases: cross-ply graphite epoxy (4) laminate with penalty factor $\lambda = 2 * 10^{10}$ and angle-ply T-graphite (5) with $\lambda = 3 * 10^{10}$. FEM validation: 30000 elements in a structured mesh, same geometry and lamination properties.

- **Configuration 6:** Two different panels are considered in this configuration. A four-ply multilayered panel is defined with two sublaminates. First sublaminate (bottom) uses an angle-ply configuration $[45, -45]^\circ$ with graphite epoxy (4) and T-graphite (5) materials, respectively. Second sublaminate (top) uses the same configuration. All plies have equal thickness and panel dimensions are $300 \times 300 \times 1mm$. The panel is fully clamped and subjected to a $1000N$ magnitude uniform load directed in the negative Z-axis direction. A uniform grid of 70×70 sampling points is employed for numerical integration, 200 neurons and *tanh* activation. Penalty factor $\lambda = 5 * 10^{10}$. FEM validation: 33000 elements in a structured mesh, same geometry and lamination properties.

  The second panel is defined in the following. An eight-ply multilayered panel is defined in three sublaminates. First sublaminate (bottom) uses an angle-ply configuration $[-45, 45]^\circ$ with graphite epoxy (4) material. Second sublaminate (middle) has four plies of T-graphite (5) in a cross-ply configuration $[0, 90, 0, 90]^\circ$. Third sublaminate (top) uses an angle-ply configuration $[45, -45]^\circ$ with graphite epoxy (4) material. All plies have equal thickness and panel dimensions are $300 \times 300 \times 2mm$. The panel is fully clamped and subjected to a $F = 1000N$ magnitude uniform load directed in the negative Z-axis direction. A uniform grid of 70×70 sampling points is employed for numerical integration, 200 neurons and *tanh* activation. Penalty factor $\lambda = 5 * 10^{10}$. FEM validation: 33000 elements in a structured mesh, same geometry and lamination properties.

- **Configuration 7:** a fully clamped graphite epoxy (4) shell under a uniform load of $1000N$ directed in the negative Z-axis direction. Longitudinal dimension is $a = 250mm$ while R, ϕ and h are free parameters. Two plies of equal thickness with cross-ply $[0, 90]^\circ$ layup sequence. Kinematic theory LD_{222} is employed, and network parameters comprehend: 70×70 sampling points, 200 neurons with *tanh* activation and penalty factor of $\lambda = 5 * 10^8$. Ritz validation: 50×50 trial functions.

- **Configuration 8:** an isotropic skew alluminum (3) plate is reported by Kale in [1], under a uniform load of $1000N$ in magnitude directed in the negative Z-axis direction. The plate is fully clamped, CCCC. Plate dimensions are $a = 1000mm$ and $h = 1mm$, dimension b is given by the aspect ratio, which is a free parameter. A skew angle, γ, is defined with respect to Y-axis and determines the inclination of side b. FEM validation: 30000 elements in a structured mesh, same geometry and material properties.

- **Configuration 9:** a circular graphite epoxy (4) plate of dimensions $R = 250mm$, $h = 1mm$ and cross-ply layup sequence $[0, 90]^\circ$ is tested in free vibrations. A second geometry considers an elliptic plate of major axis $a = 250mm$ and minor axis $b = 125mm$ of the same material and layup configuration. A third geometry considers an elliptic plate of major axis $a = 250mm$ and minor axis $b = 62.5mm$ of the same material and layup configuration. LD_{110} kinematic theory along with CCCC boundary conditions are used. Network parameters: 100×100 sampling points for numerical integration, 200 neurons with *tanh* activation and modified range of weights $w = [-3, 3]$. Penalty factor $\lambda = 10^9$. FEM validation: 28000 square elements in unstructured mesh.

FEM validation is generally used when composites with multiple layers are considered, in order to take advantage of the modeling properties of ACP module. It has been also used whenever a first comparison of the natural frequencies in free vibrations is required, or whenever Ritz approximation is not possible due to code limitations such as plates with non-rectangular shapes. Ritz solutions are used to compare the majority of results, the code has been extensively validated by previous authors and can be taken as reference. This criteria is motivated by the extremely fast execution of the Ritz method compared with FEM. A relative error of 5% is considered an acceptable threshold both for displacements or stresses when comparisons are made with any of these tools, it is used as a default criteria. FEM solutions are by definition not exact, and the nature of the network approximation makes the result to be a prediction subjected to certain accuracy errors.

5.2 Strong-form collocation method - validation

Cases of study for the strong-form formulation are focused on pure bending problems only. The external load introduced in the structure can be considered of any kind, from uniform to concentrated loads, but restricted to the transverse direction. This first collocation method has been validated both in the context of linear static analysis and free vibrations, for which different study cases are reported.

Convergence study - linear static analysis

It is important to determine the convergence rate of the displacements regarding the order of the kinematic expansion used for the approximation, or if on the contrary, a certain deviation is expected for higher order theories. *Configuration 1* is used to evaluate the transverse displacement at the point of maximum for different order kinematic theories and sets of boundary conditions. The evaluation is shown in Fig.(15).

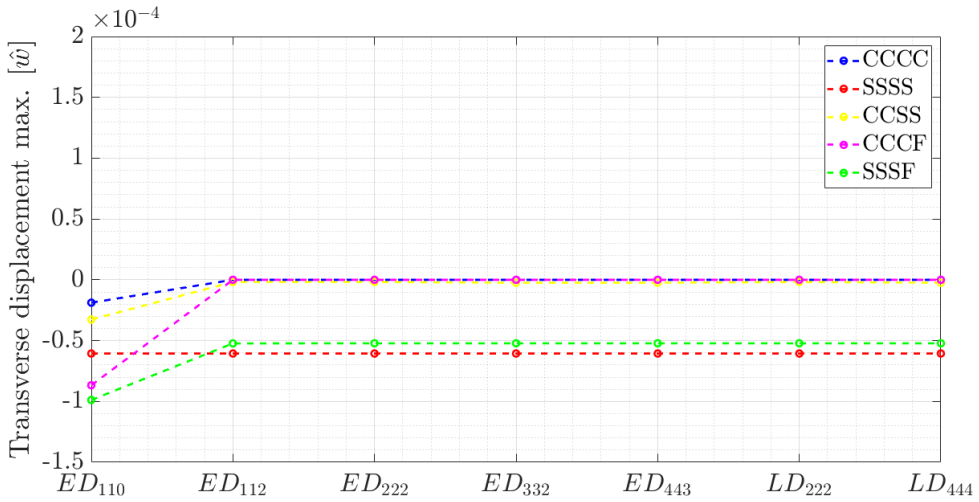


Figure 15: Convergence of the first collocation method on different boundary conditions with respect to kinematic theories. Case setup from *Configuration 1*.

Early from ED_{112} model, all theories converge to the same respective values of the transverse displacement for every set of boundary conditions. In Table [5] the previous results are compared with the Ritz method using the same models.

Case	**Model**	**CCCC [m]**	**SSSS [m]**	**CCSS [m]**	**CCCF [m]**	**SSSF [m]**
Present	ED_{110}	1.88 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	8.67 x 10^{-5}	9.88 x 10^{-5}
	ED_{112}	9.70 x 10^{-10}	6.06 x 10^{-5}	1.77 x 10^{-6}	7.83 x 10^{-10}	5.23 x 10^{-5}
	ED_{222}	9.70 x 10^{-10}	6.06 x 10^{-5}	1.77 x 10^{-6}	8.01 x 10^{-10}	5.22 x 10^{-5}
	ED_{332}	8.31 x 10^{-10}	6.06 x 10^{-5}	2.41 x 10^{-6}	8.21 x 10^{-10}	5.22 x 10^{-5}
	ED_{443}	8.31 x 10^{-10}	6.06 x 10^{-5}	2.40 x 10^{-6}	8.22 x 10^{-10}	5.22 x 10^{-5}
	LD_{222}	9.70 x 10^{-10}	6.06 x 10^{-5}	1.77 x 10^{-6}	8.01 x 10^{-10}	5.22 x 10^{-5}
	LD_{444}	8.34 x 10^{-10}	6.06 x 10^{-5}	2.41 x 10^{-6}	8.10 x 10^{-10}	5.22 x 10^{-5}
Ritz	ED_{110}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.28 x 10^{-5}	4.50 x 10^{-5}	1.97 x 10^{-4}
	ED_{112}	1.88 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{222}	1.88 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{332}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{443}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	LD_{222}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	LD_{444}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}

Table 5: Comparison of different high order variable-kinematic theories and sets of boundary conditions. Ritz validation: 50×50 trial functions. Case setup from *Configuration 1*.

It can be seen that the obtained results only approximate to the reference values with the first theory, except for the fully simply-supported case that behaves well for any expansion order. It is possible to clearly state that there are numerical issues regarding high order theories and free edges boundary conditions for the strong formulation method. Finding a solution to these problems will be left for future work. For this reason, in the remaining part of the thesis when using the strong formulation only the ED_{110} kinematic theory and no free edge boundary conditions will be considered.

Case study 1 - alluminum plate of different geometries

Due to the large amount of variations introduced in the same case geometry, the validation is done against a Ritz implementation with the same conditions. *Configuration 1* is used as baseline, and its dimensions are modified from those of a thin square plate to modified thickness and aspect ratio. Table [6] reports two different locations where the transverse displacement is evaluated for every configuration. Three different aspect ratios (a/b) and thickness ratios (a/h) are considered in order to span a range of configurations that involves both slender and square geometries, as well as thin and thick plates. CCCC boundary conditions and ED_{110} theory are maintained for all cases.

a/b	**a/h**	**x=**	**y=**	**Present [m]**	**Ritz [m]**
0.5	1	0	0	5.54×10^{-12}	5.54×10^{-12}
		$a/4$	$b/4$	3.53×10^{-12}	3.48×10^{-12}
	10	0	0	4.34×10^{-9}	4.34×10^{-9}
		$a/4$	$b/4$	2.05×10^{-9}	2.00×10^{-9}
	100	0	0	3.77×10^{-5}	3.78×10^{-5}
		$a/4$	$b/4$	1.76×10^{-5}	1.65×10^{-5}
1	1	0	0	3.49×10^{-12}	3.49×10^{-12}
		$a/4$	$b/4$	2.19×10^{-12}	2.09×10^{-12}
	10	0	0	2.26×10^{-9}	2.26×10^{-9}
		$a/4$	$b/4$	9.33×10^{-10}	9.06×10^{-10}
	100	0	0	1.88×10^{-5}	1.89×10^{-5}
		$a/4$	$b/4$	7.19×10^{-6}	6.89×10^{-6}
5	1	0	0	2.23×10^{-13}	2.23×10^{-13}
		$a/4$	$b/4$	1.69×10^{-13}	1.64×10^{-13}
	10	0	0	2.85×10^{-11}	2.85×10^{-11}
		$a/4$	$b/4$	2.08×10^{-11}	2.02×10^{-11}
	100	0	0	6.42×10^{-8}	6.43×10^{-8}
		$a/4$	$b/4$	3.86×10^{-8}	3.67×10^{-8}

Table 6: Validation of an alluminum plate with different relative geometries (a/b, a/h). Case setup from *Configuration 1*.

The results show a good performance in terms of prediction of transverse displacements is reported for all tested configurations. Slightly larger differences are reported at point $(a/4, b/4)$, effect that could be introduced by the different interpolation grid used in both methods, although it does not affect greatly the quality of the solution compared to the maximum transverse displacement point.

Case study 2 - alluminum plate under different types of transverse load

Four different transverse loads are used in the plate reported by *Configuration 1*, and considering ED_{110} as the kinematic theory along with fully simply-supported boundaries (SSSS). The motivation of this test is to show how well the method performs when the load is not uniformly distributed, but zero load is present at part of the collocation points. This fact introduces a discontinuity of the load from point to point, what makes it an interesting feature in validation. Uniform and bisinusoidal loads are also compared, and Ritz validation (50×50 trial functions) is used to obtain the values of the transverse displacement evaluated at the points of maximum

deflection. This comparison is reported in Table [7] with two localised loads applied in a square centered at the origin of the plate.

Type	Case	Value [m]
Uniform	Present	-6.06×10^{-5}
	Ritz	-6.06×10^{-5}
Bi-sinusoidal	Present	-3.93×10^{-5}
	Ritz	-3.83×10^{-5}
Localised $(50 \times 50mm)$	Present	-2.98×10^{-5}
	Ritz	-3.18×10^{-5}
Localised $(20 \times 20mm)$	Present	-4.53×10^{-6}
	Ritz	-6.49×10^{-6}

Table 7: Validation of different transverse load types on a SSSS alluminum (1) square plate. Case setup from *Configuration 1*.

A sketch of the way these loads are introduced into the plate acting in the transverse direction is represented in Fig.(16).

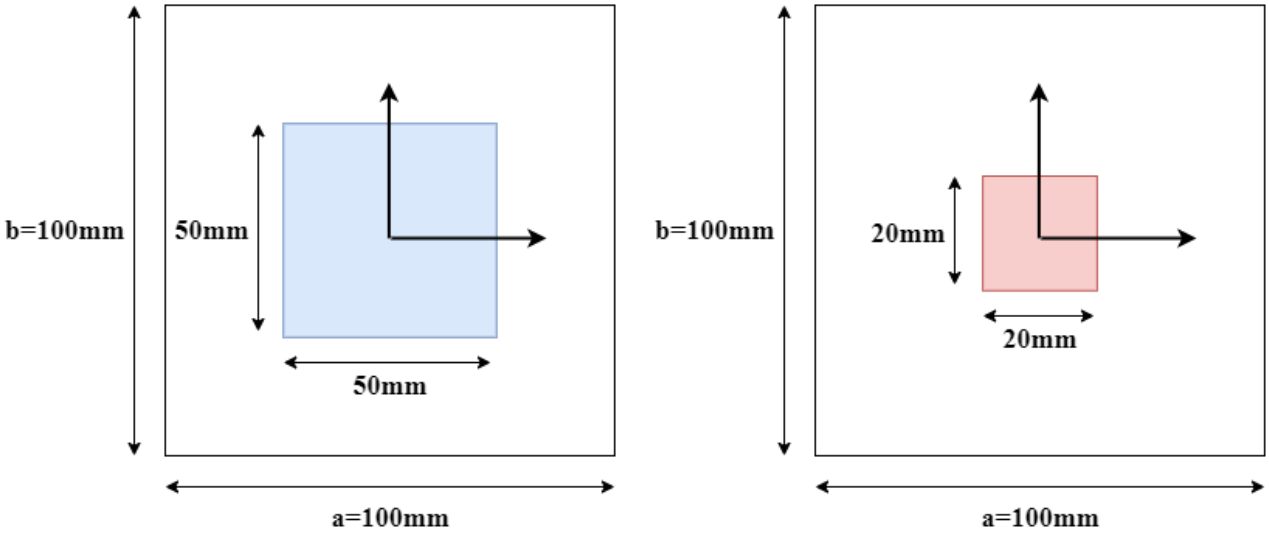


Figure 16: Sketch of the area of application of loads for the validation case of localised loads in Table [7]. $50 \times 50mm$ centered area (*left*), $20 \times 20mm$ centered area (*right*).

The results show an agreement in the prediction of displacements with both methods, that worsens with the reduction of the area of application of the load. Uniform and bisinusoidal loads are well predicted with the collocation method, but a large difference is observed in the other cases, that are underestimated with respect to reference. In view of the results, only distributed loads are considered to pass the validation and could be applied to other cases under consideration.

A comparison of the transverse displacement contours of the plate under localised loads is shown in Fig.(17)-(18).

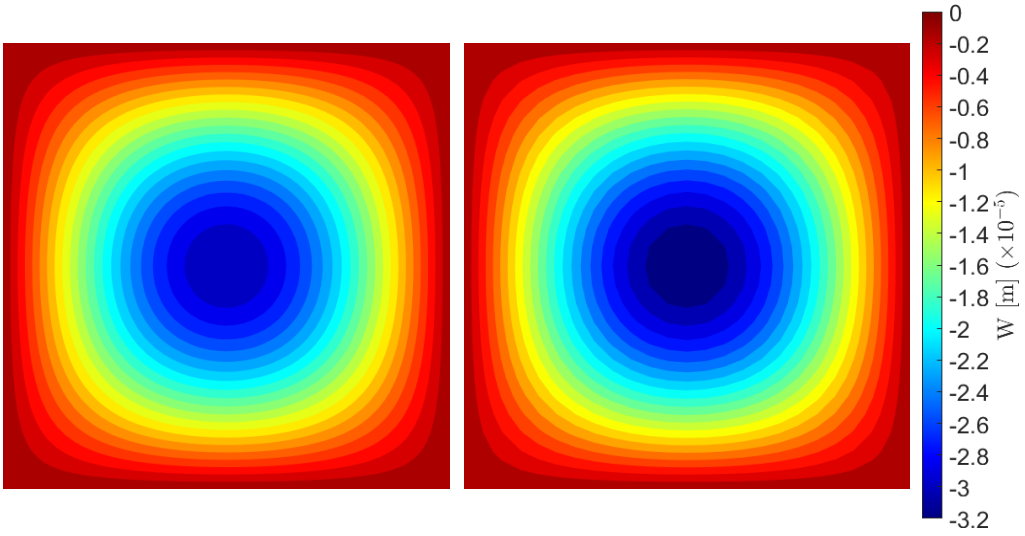


Figure 17: Contours of transverse displacement for the case of a localised load in a 50×50 square area. Contours: collocation method (*left*) and Ritz method (*right*). Case setup from *Configuration 1*.

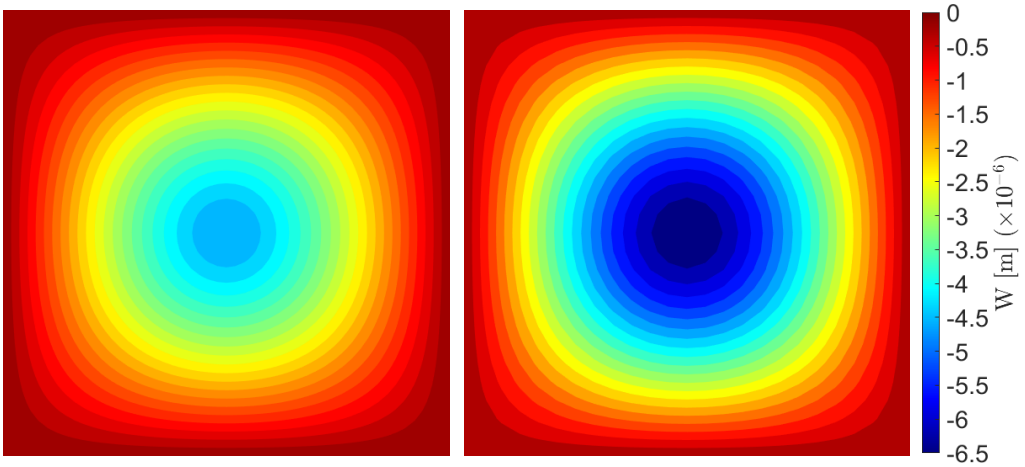


Figure 18: Contours of transverse displacement for the case of a localised load in a 20×20 square area. Contours: collocation method (*left*) and Ritz method (*right*). Case setup from *Configuration 1*.

The shape of the contour coincides with the reference one, but the magnitude becomes progressively underestimated the smaller the application area considered.

Case study 3 - multilayered composite plate with cross-ply and angle-ply configurations

Bending analysis of multilayered plates makes the main target of this implementation. The composite plate in *Configuration 2* is tested for different materials and layup sequence combinations. In the first configuration, the plate is made by three plies of *graphite epoxy* (4) in cross-ply sequence, while in the second case, the material is changed to *T-graphite* (5), they both reported by Fantuzzi [2]. This second case uses an angle-ply sequence. Table [8] introduces the validation of these configurations against the same solution given by a FEM simulation. The setup uses a structured mesh of 30000 elements. The value of the transverse displacement

is given at three different locations relative to the dimensions of the plate and for three different sets of boundary conditions, in order to extend the comparison.

Layup	Boundary conditions	x=	y=	FEM [m]	Present [m]
$[0, 90, 0]°(4)$	CCCC	$a/4$	$b/4$	-9.05 x 10^{-5}	-9.05 x 10^{-5}
		$-a/4$	$-b/4$	-9.05 x 10^{-5}	-9.05 x 10^{-5}
		$a/2$	$b/2$	-2.27 x 10^{-4}	-2.27 x 10^{-4}
	SSSS	$a/4$	$b/4$	-5.39 x 10^{-4}	-5.38 x 10^{-4}
		$-a/4$	$-b/4$	-5.39 x 10^{-4}	-5.38 x 10^{-4}
		$a/2$	$b/2$	-10 x 10^{-4}	-10 x 10^{-4}
	CCSS	$a/4$	$b/4$	-2.18 x 10^{-4}	-2.18 x 10^{-4}
		$-a/4$	$-b/4$	-1.74 x 10^{-4}	-1.74 x 10^{-4}
		$a/2$	$b/2$	-4.31 x 10^{-4}	-4.30 x 10^{-4}
$[-45, 0, 45]°(5)$	CCCC	$a/4$	$b/4$	-6.38 x 10^{-5}	-6.18 x 10^{-5}
		$-a/4$	$-b/4$	-6.38 x 10^{-5}	-6.18 x 10^{-5}
		$a/2$	$b/2$	-1.30 x 10^{-4}	-1.28 x 10^{-4}
	SSSS	$a/4$	$b/4$	-1.86 x 10^{-4}	-1.76 x 10^{-4}
		$-a/4$	$-b/4$	-1.86 x 10^{-4}	-1.76 x 10^{-4}
		$a/2$	$b/2$	-3.56 x 10^{-4}	-3.39 x 10^{-4}
	CCSS	$a/4$	$b/4$	-1.35 x 10^{-4}	-1.20 x 10^{-4}
		$-a/4$	$-b/4$	-9.29 x 10^{-5}	-7.50 x 10^{-5}
		$a/2$	$b/2$	-2.15 x 10^{-4}	-1.85 x 10^{-4}

Table 8: Layup sequences and composite materials for different boundary conditions. Transverse displacement evaluated at midsurface. Case setup from *Configuration 2*.

Overall, the prediction matches well the FEM results for the first sequence in all tested boundary conditions. A slight worsening is predicted for the cross-ply case, generally in the form of an underestimation. It is interesting to note how the method finds it more difficult to approximate the behaviour of a plate with a configuration that deviates from cross-ply. Additional cases of a general layup show the same behaviour, and can be found in Appendix B. This test case has been limited to the use of a first-order ED_{110} theory, even though the use of a higher order theory and specifically layerwise theories might improve the solution field in a laminate due to the capacity to represent the heterogeneity of the layup. This option is prevented due to the convergence problems previously reported, limiting the extent of the analysis.

Convergence study - free vibrations analysis

An additional study for the convergence of natural frequencies is required in the context of free vibrations. The same *Configuration 1* is simulated with different kinematic theories to recover a subset of natural frequencies used in validation with Ritz results. A modification is introduced in the number of hidden nodes, now reduced to 100, and a pseudoinverse preconditioner matrix is used. The evolution of the first three natural frequencies for different sets of boundary conditions is shown in Figure (19).

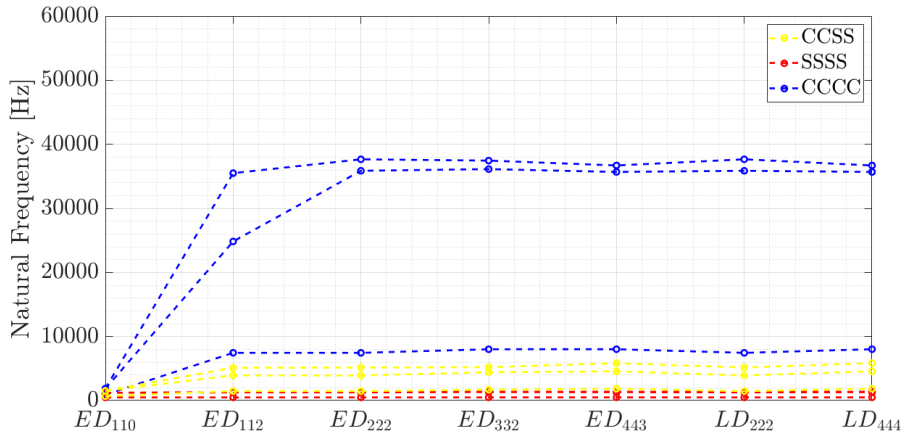


Figure 19: Convergence of the first collocation method in free vibrations with respect to kinematic theory. CCCC, SSSS and CCSS boundary conditions are considered. Case setup from *Configuration 1*.

In the first case, the values of the natural frequencies do not exactly converge towards high order theories, but stay in an oscillation behaviour. From theory ED_{222} this behaviour is reported in all sets of boundaries, with the exception of SSSS that presents an asymptotic behaviour early from FSDT theory.

A comparison is done regarding the same implementation in the Ritz method, reported in Table [9]. The frequencies within theory ED_{222} are taken for reference regarding the previous results.

BC	**Case**	**Mode [Hz]**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CCCC	Present	7446.22	35863.01	37654.19	42311.93	46737.56	49056.12	52442.68	54330.36	56090.54	68044.91
	Ritz	880.34	1793.82	1793.82	2642.20	3211.71	3227.18	4023.45	4023.45	5130.09	5130.09
SSSS	Present	489.97	1269.61	1336.40	2273.71	2796.37	3656.74	4976.22	8100.80	8426.31	15415.21
	Ritz	483.22	1207.53	1207.53	1930.72	2413.02	2413.02	3134.64	3134.64	4096.95	4096.95
CCSS	Present	1475.98	3911.74	5153.33	12041.82	18791.24	19400.82	31892.81	32202.28	39125.39	45871.82
	Ritz	662.10	1480.52	1486.62	2268.46	2798.61	2802.27	3558.24	3565.69	4597.32	4599.61

Table 9: First ten natural frequencies compared to Ritz method. Kinematic theory ED_{222}. Case setup from *Configuration 1*. BC: boundary conditions.

A large disagreement is observed inside every case with reference results. In a general sense, the values of the natural frequencies are largely overpredicted by the collocation method. The only relative improvement can be considered for SSSS case in modes 1,2 where this difference is not as large as in the others. By recalling the results in Fig.(19), it has been said that a complete convergence is not reached, but these frequencies are also far away from reference cases. It is for ED_{110} theory where natural frequencies are closer to reference, and the method fails to extend the predictions to higher order theories. This is the same incorrect prediction occurring for the displacements with other theories rather than FSDT in the context of the linear static analysis. Therefore, this theory can be considered the only one that approximates to reference in an acceptable way, and the analysis will be restricted to its use.

Case study 4 - alluminum plate of different geometries in free vibrations

In order to better understand the capabilities of the free vibrations solver, plate *Configuration 1* is simulated for different plate geometries, in the same way of what is shown for *Case study 1*. Due to the fact that the method has shown a good applicability for a square plate,

only those more extremal cases are considered. The idea is to identify if there is a performance drop for slender and thicker plates and if the found modes correspond to those reported by the Ritz method.

Table [10] reports the first six natural frequencies found with both solution methods at four different configurations.

a/b	a/h	**Case**	**Mode [Hz]**					
			1	**2**	**3**	**4**	**5**	**6**
0.5	10	Present	55808.13	71474.95	98675.78	133994.49	136992.12	147409.43
		Ritz	55662.17	71124.70	98025.64	133957.89	134908.63	147256.05
	100	Present	612.81	952.69	1474.59	1580.64	1865.14	3121.75
		Ritz	601.54	778.71	1095.09	1548.56	1564.45	1737.39
5	10	Present	629805.70	649898.46	687748.27	745065.60	811678.18	843704.36
		Ritz	629718.37	649835.07	687758.04	745226.59	811833.05	821182.01
	100	Present	13594.56	14116.42	15202.92	20051.76	36372.65	36986.86
		Ritz	13591.83	14065.83	14907.65	16174.60	17917.08	20167.32

Table 10: First six natural frequencies of a plate for different relative geometry configurations. Case setup from *Configuration 1*.

It is important to remark the large difference in magnitude of the natural frequencies between thick and thin configurations. It is one of the motivations of this analysis to test how different these frequencies are among them and if the prediction improves for thicker plates. In the first case ($a/b = 0.5, a/h = 10$), the thickness of the plate with respect to planar dimensions shifts the natural modes to high frequencies, where the reported agreement with Ritz solutions is good. The difference in tenths or hundreds of Hz is inherent to the nature of the method, that finds it difficult to match the exact Ritz frequency, however, it corresponds to a percentual difference on the order of 1.5% at the worst case (mode 5), well within tolerance limits. The contours of transverse displacement are reported in Fig.(20) for this particular case, where the contours of the first modes appear in a clean fashion, well represented by the method. The reasoning is extensible also to the third configuration ($a/b = 5, a/h = 10$), with a larger difference reported however for the sixth mode. There is a change of tendency by considering thin plate configurations, similar frequencies are reported for the first or two modes, but the similarity weakens in the following ones. The collocation method starts to overpredict the natural frequencies, that easily go beyond accepted tolerance. It is possible to conclude that the collocation method finds an increased difficulty in solving the free vibrations problem for thin plates, while the prediction on thick configurations give accurate results. The impossibility to perform further tests with higher order theories hinders the applicability of the method in this analysis.

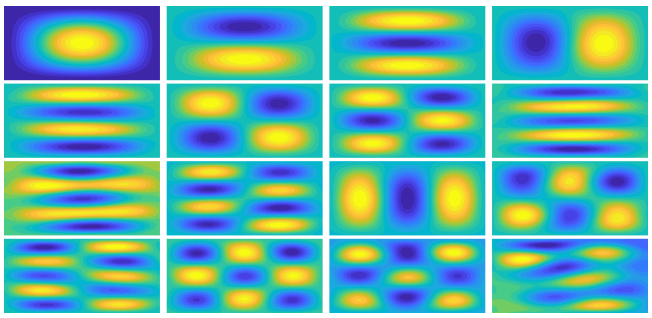


Figure 20: Contours of transverse displacements for the first 16 modes of an alluminum plate of thickness $h = 1mm$ and relative dimensions $a/b = 0.5$, $a/h = 10$. Case setup from *Configuration 1*.

Case study 5 - cross-ply laminate plate with different boundary conditions

The following test considers the plate of *Configuration 3*. It uses a specific composite material reported by Pagano [66], with strongly marked orthotropic properties. Young's modulus is of two orders of magnitude larger in the longitudinal direction of the fibers, and shear modulus are considerably lower than those found in alluminum alloys. The reason to validate this case resides on the extension of the comparison in free vibrations to deal with a laminate, in order to assess the performance in predicting the natural frequencies in a layup where orthotropy plays a primary role, rather than the isotropic cases tested so far. Different sets of boundary conditions, namely CCCC, SSSS and CCSS are considered. Validation is performed against a FEM simulation as reported by *Configuration 3*.

Table [11] reports the first six natural frequencies obtained in the different configurations, both with ED_{110} and LD_{110} theories, and compared with FEM results.

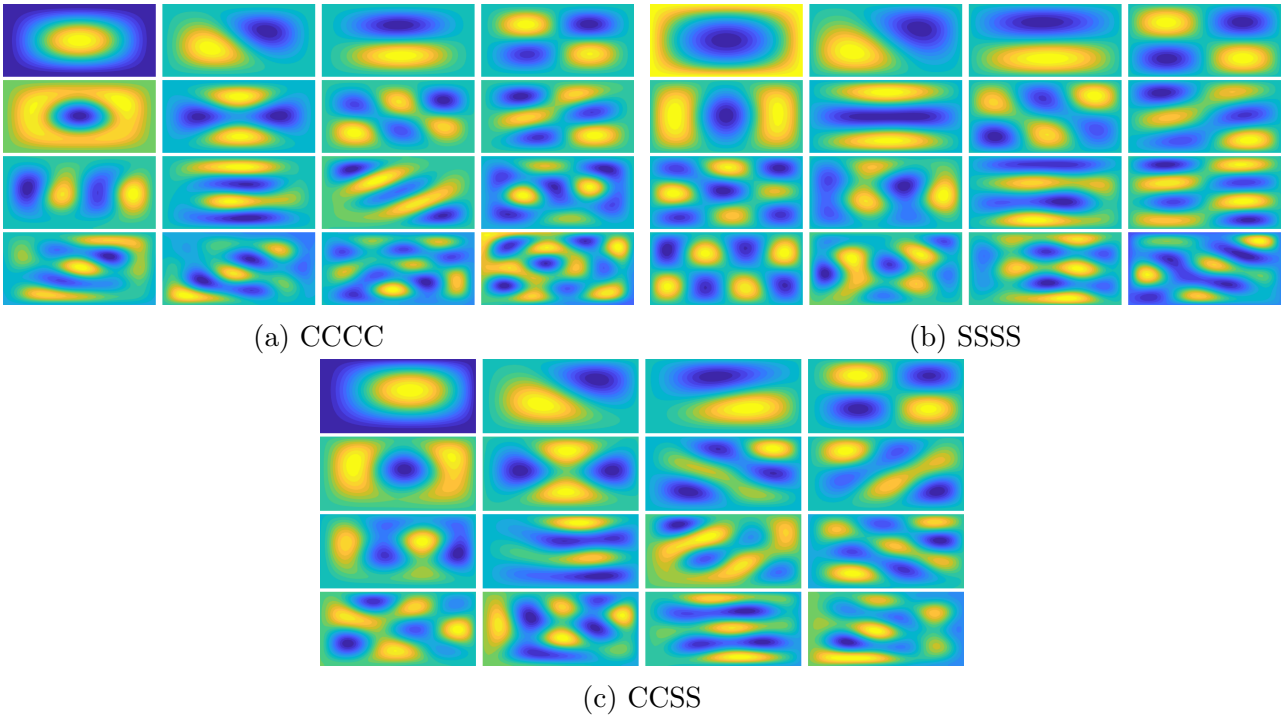
Boundary conditions	**Mode**	**Present - ESL [Hz]**	**Present - LW [Hz]**	**FEM [Hz]**
CCCC	1	704.85	703.99	700.14
	2	2038.75	1455.29	1440.37
	3	2701.53	1466.34	1440.37
	4	2712.82	2100.42	1992.80
	5	3408.57	2716.13	2647.56
	6	3614.32	2813.77	2654.85
SSSS	1	339.06	338.98	339.07
	2	922.77	922.30	922.69
	3	923.35	923.19	922.69
	4	1355.24	1358.84	1354.01
	5	1967.53	1987.86	1951.52
	6	1978.80	2017.99	1951.56
CCSS	1	501.71	503.91	500.04
	2	1167.30	1174.05	1163.76
	3	1168.94	1177.04	1166.89
	4	1671.57	1696.43	1656.02
	5	2319.13	2362.66	2286.37
	6	2326.42	2386.05	2287.94

Table 11: First six natural frequencies obtained in different configurations on a cross-ply two-ply composite plate. Validation performed with a FEM modal analysis. Case setup from *Configuration 3*.

The implementation of a layerwise theory in free vibrations makes the element of novelty of this test case. LW theories have been found to be subjected to the same convergence problems

reported in ESL high order theories, with the sole exception of LD_{110}. This is an interesting point, considering this same theory could not be applied in the context of linear static composite analysis in *Case study 3*, due to a completely erroneous prediction of the displacement field. Regarding ESL cases validation, results shows a good prediction, except for the fully clamped plate. These results confirm what was reported in Table [24], the collocation method has difficulties in finding the natural frequencies when clamped edges are present compared to others more loosely constraint. SSSS errors are minimum, while slightly higher in CCSS configuration. This is a set of mixed conditions, and it still performs within admissible error range (5%). However, with a CCCC plate natural modes are easily missed during the computation and the frequencies reported are largely overpredicted. This issue does not obey to any special consideration and was not expected to happen, the imposition of essential conditions regarding the strong formulation is done in a straightforward manner with additional equations that limit the displacement at the boundary. Regarding LW implementation, all cases show an agreement within tolerance limits, including CCCC case, which frequencies are now much more approximated to reference ones. Furthermore, the method has shown a good performance dealing with orthotropic materials in cross-ply, and can be possibly extended to other configurations where a larger number of plies or different materials are considered. As a conclusion, this test case opens the possibility to use either ED_{110} of LD_{110} theories when it comes to modal analysis, and confirms the same performance reported in isotropic plates as well as its limitations (free edges).

The contours of transverse displacement of the LW theory test cases are reported in Fig.(21).



(a) CCCC (b) SSSS

(c) CCSS

Figure 21: Contours of transverse displacement for the first 16 modes of a cross-ply composite plate. Three configurations: a) CCCC, b) SSSS and c) CCSS. Case setup from *Configuration 3*.

The strong-form formulation has been analyzed from the perspective of different configurations aimed to test its performance. Both isotropic/orthotropic materials, varying geometries and load conditions have proven the efficacy of the collocation method ruled by a PINN-ELM framework, which has shown a limited performance. Two major problems are reported to affect the method in general terms, both for a linear static and free vibrations analysis: simulation of

higher order kinematic theories above FSDT, and dealing with the imposition of free edges in the panel. Furthermore, the influence of different neural network parameters over the solution remains to be tested. Both these issues are widely covered in Section [5.4] with a dedicated discussion.

5.3 Energy method - validation

The weak formulation constitutes an alternative strategy that has been considered, where pure bending problems are solved for a selection of reference configurations as well. The methodology for the implementation of the weak-form collocation method was extensively introduced in Section [3.5.2]. Some conclusions belonging to the first part of this discussion on the strong formulation method are applicable to the energy method and therefore are omitted. They are referred to the validation of the method for different relative geometries and types of load. Those analyses showed the method can obtain good performance when different geometries, either thin/thick and slender plates are simulated, both in terms of the transverse displacement field and the frequencies associated to the natural modes in free vibrations. A drop in performance was reported for thin plates in free vibrations, argument that holds in this method. With respect to different transverse loads, the same problematic is expected when localised loads are applied, while uniform loads are expected to work fine in every case.

Convergence study - linear static analysis

With respect to the order of the kinematic expansion, this first study uses *Configuration 4* with different kinematic theories, to test the rate of convergence at the maximum transverse displacement. Different sets of boundary conditions are used, and the evaluation is reported in Fig.(22).

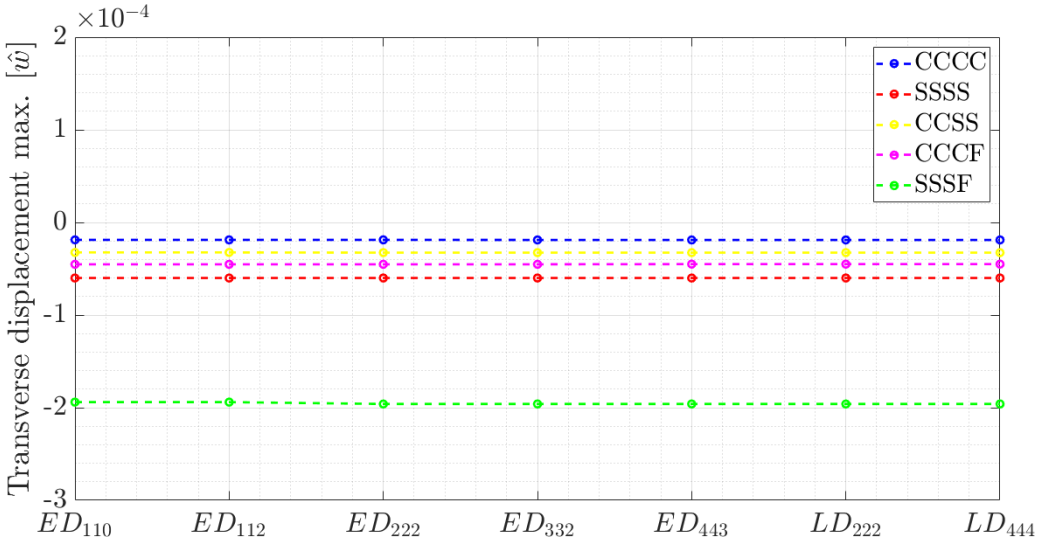


Figure 22: Convergence study of the maximum transverse displacement with different kinematic theories. Case setup from *Configuration 4*.

An almost immediate convergence can be observed for all test cases. The numerical values for the displacements are validated with their counterpart obtained in a Ritz implementation. The results are tabulated in Table [12] for the same conditions reported in Fig.(22).

Case	Model	CCCC [m]	SSSS [m]	CCSS [m]	CCCF [m]	SSSF [m]
Present ($\lambda = 10^{10}$)	ED_{110}	1.88 x 10^{-5}	6.00 x 10^{-5}	3.24 x 10^{-5}	4.52 x 10^{-5}	1.94 x 10^{-4}
	ED_{112}	1.88 x 10^{-5}	6.00 x 10^{-5}	3.24 x 10^{-5}	4.52 x 10^{-5}	1.94 x 10^{-4}
	ED_{222}	1.89 x 10^{-5}	6.00 x 10^{-5}	3.26 x 10^{-5}	4.52 x 10^{-5}	1.96 x 10^{-4}
	ED_{332}	1.89 x 10^{-5}	6.01 x 10^{-5}	3.26 x 10^{-5}	4.52 x 10^{-5}	1.96 x 10^{-4}
	LD_{443}	1.89 x 10^{-5}	6.01 x 10^{-5}	3.26 x 10^{-5}	4.50 x 10^{-5}	1.96 x 10^{-4}
	LD_{222}	1.89 x 10^{-5}	6.01 x 10^{-5}	3.26 x 10^{-5}	4.50 x 10^{-5}	1.96 x 10^{-4}
	LD_{444}	1.89 x 10^{-5}	6.01 x 10^{-5}	3.26 x 10^{-5}	4.50 x 10^{-5}	1.96 x 10^{-4}
Ritz	ED_{110}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.28 x 10^{-5}	4.50 x 10^{-5}	1.97 x 10^{-4}
	ED_{112}	1.88 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{222}	1.88 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{332}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	ED_{443}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	LD_{222}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}
	LD_{444}	1.89 x 10^{-5}	6.06 x 10^{-5}	3.27 x 10^{-5}	4.49 x 10^{-5}	1.97 x 10^{-4}

Table 12: Comparison of different high order variable-kinematic theories and sets of boundary conditions. Ritz validation: 50×50 trial functions. Case setup from *Configuration 4*.

These results are in agreement with the motivation to shift to this new method. The problem regarding the introduction of free edges is avoided thanks to the enforcement of natural boundary conditions inherent of the energy formulation. Moreover, all kinematic theories show an almost equal performance that tends to the reference values, with a minimal error well within tolerance (5%). This test case opens the possibility to simulate more complicated setups, a topic deeply covered in the next study cases.

Case study 6 - multilayered composite plate with different layup sequences and materials

An extensive comparison is done regarding the use of composite multilayered plates with the setup of *Configuration 5*, both in cross-ply and angle-ply layup sequences. Two different materials are used: *graphite epoxy* (4) and *T-graphite* (5). The possibility to use higher order theories is now available, and in particular LW ones are a relevant choice for a multilayered panel. LD_{110} is used along with theory LD_{222} in order to compare their performance, which includes a higher order description of the trasverse displacement. Both configurations are validated as well in a FEM simulation, and different sets of boundary conditions are used to study any special effect that could be introduced by them.

Complete results are reported in Table [13], where the initialization value of the penalty factor λ is reported as well in each subcase for reproducibility.

Layup	Boundary conditions	x=	y=	FEM [m]	Present - LD_{110} [m]	Present - LD_{222} [m]
$[0, 90, 0]^\circ$ (4)	CCCC ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-9.76 x 10^{-6}	-9.43 x 10^{-6}	-9.36 x 10^{-6}
		$-a/4$	$-b/4$	-9.76 x 10^{-6}	-9.45 x 10^{-6}	-9.38 x 10^{-6}
		$a/2$	$b/2$	-2.40 x 10^{-5}	-2.34 x 10^{-5}	-2.36 x 10^{-5}
	SSSS ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-5.74 x 10^{-5}	-5.68 x 10^{-5}	-5.67 x 10^{-5}
		$-a/4$	$-b/4$	-5.74 x 10^{-5}	-5.68 x 10^{-5}	-5.68 x 10^{-5}
		$a/2$	$b/2$	-1.06 x 10^{-4}	-1.05 x 10^{-4}	-1.05 x 10^{-4}
	CCSS ($\lambda = 3 * 10^{10}$)	$a/4$	$b/4$	-1.86 x 10^{-5}	-1.83 x 10^{-5}	-1.83 x 10^{-5}
		$-a/4$	$-b/4$	-2.35 x 10^{-5}	-2.28 x 10^{-5}	-2.23 x 10^{-5}
		$a/2$	$b/2$	-4.55 x 10^{-5}	-4.48 x 10^{-5}	-4.53 x 10^{-5}
	CCFF ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-9.75 x 10^{-5}	-9.65 x 10^{-5}	-9.40 x 10^{-6}
		$-a/4$	$-b/4$	-1.59 x 10^{-4}	-1.53 x 10^{-4}	-1.35 x 10^{-4}
		$a/2$	$b/2$	-2.20 x 10^{-4}	-2.18 x 10^{-4}	-2.26 x 10^{-4}
	SSSF ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-8.42 x 10^{-5}	-8.41 x 10^{-5}	-8.40 x 10^{-5}
		$-a/4$	$-b/4$	-5.66 x 10^{-5}	-5.61 x 10^{-5}	-5.61 x 10^{-5}
		$a/2$	$b/2$	-1.11 x 10^{-4}	-1.10 x 10^{-4}	-1.10 x 10^{-4}
$[-45, 0, 45]^\circ$ (5)	CCCC ($\lambda = 3 * 10^{10}$)	$a/4$	$b/4$	-1.78 x 10^{-5}	-1.68 x 10^{-5}	-1.72 x 10^{-5}
		$-a/4$	$-b/4$	-1.78 x 10^{-5}	-1.68 x 10^{-5}	-1.73 x 10^{-5}
		$a/2$	$b/2$	-4.66 x 10^{-5}	-4.49 x 10^{-5}	-4.63 x 10^{-5}
	SSSS ($\lambda = 3 * 10^{10}$)	$a/4$	$b/4$	-4.33 x 10^{-5}	-4.21 x 10^{-5}	-4.17 x 10^{-5}
		$-a/4$	$-b/4$	-4.33 x 10^{-5}	-4.21 x 10^{-5}	-4.20x 10^{-5}
		$a/2$	$b/2$	-9.07 x 10^{-5}	-8.84 x 10^{-5}	-8.85 x 10^{-5}
	CCSS ($\lambda = 10^{10}$)	$a/4$	$b/4$	-2.84 x 10^{-5}	-2.80 x 10^{-5}	-2.80 x 10^{-5}
		$-a/4$	$-b/4$	-2.80 x 10^{-5}	-2.76 x 10^{-5}	-2.79 x 10^{-5}
		$a/2$	$b/2$	-6.58 x 10^{-5}	-6.50 x 10^{-5}	-6.49 x 10^{-5}
	CCFF ($\lambda = 10^{10}$)	$a/4$	$b/4$	-1.77 x 10^{-4}	-1.76 x 10^{-4}	-1.76 x 10^{-4}
		$-a/4$	$-b/4$	-1.86 x 10^{-4}	-1.85 x 10^{-4}	-1.85 x 10^{-4}
		$a/2$	$b/2$	-3.15 x 10^{-4}	-3.16 x 10^{-4}	-3.17 x 10^{-4}
	SSSF ($\lambda = 5 * 10^{9}$)	$a/4$	$b/4$	-1.17 x 10^{-4}	-1.10 x 10^{-4}	-1.14x 10^{-4}
		$-a/4$	$-b/4$	-2.77 x 10^{-4}	-2.37 x 10^{-4}	-2.58 x 10^{-4}
		$a/2$	$b/2$	-2.95 x 10^{-4}	-2.73 x 10^{-4}	-2.86 x 10^{-4}

Table 13: Layup sequences and composite materials for two different configurations - cross-ply and angle-ply. Transverse displacement evaluated at three points in the domain. Validation against FEM simulation results. Case setup from *Configuration 5*.

The agreement with reference solutions is exceptionally good in both theories. No differences (above error tolerance) are reported and it is possible to conclude that this second method presents an even better performance to reproduce displacement fields in multilayered composites, than its strong method counterpart. Both theories present very similar results, therefore LD_{110} can be used to sufficiently describe the problem with a lower computational cost. Next step would be to test if this performance is maintained when the assembly of different materials and sublaminates are considered. The value of the penalty factor has been slightly modified according to the type of boundary conditions introduced. Experience has shown that a lower value of the penalty tends to give better results in angle-ply situations or when free edges are introduced in the setup. The latter conclusion is expected to be related with a softly constraint problem at matricial level, that requires of less penalty components present in the network approximation. This is an expected tendency, but not a rule and some problems disagree, especially when different boundary conditions are mixed.

Case study 7 - multilayered composite plate with assembly of multiple sublaminates

Another test is presented, focused on reporting the method performance to the assembly of several sublaminates. This assembly process is always treated in a layerwise manner, in order to reflect the different material properties and fiber orientations regarding that sublaminates are a priori stablished to model different regions of the structure. The two panels of *Configuration 6* are used to exemplify cases with multiple layers divided in different sublaminates. The

first layup contemplates four plies of equal-thickness divided in two sublaminates, in both cases using an angle-ply sequence. Its motivation is to show how well the method is capable of reproducing the change in material properties between sublaminates. A second case considers eight plies of equal thickness divided in three sublaminates, top and bottom ones use an angle-ply sequence and are made of *graphite epoxy* (4) while the central sublaminate occupies four plies in cross-ply configuration. The objective is to analyze how well the method simulates a change in the layup sequence between sublaminates. In both cases, a plate of $300 \times 300mm$ dimensions is used, bigger than the plate used in previous configurations. Thicknesses are $h = 1mm$ and $h = 2mm$, and together with plate dimensions make a ratio of $a/h = 300$ and $a/h = 150$ which have been considered good examples of a thin plate. Collocation grid size has been reduced to 70×70 points to deal with the large computational requirements. Regarding the experience from *Case study 7*, LD_{110} is considered a sufficiently accurate theory for modelling. It is used alone in the first case to simulate both sublaminates. The second configuration is modelled as $LD_{110}/ED_{222}/LD_{110}$, aimed to also validate for the first time the combination of LW/ESL theories, a unique feature of SGUF.

Table [14] includes several measurements of the transverse displacement at different locations for both cases of interest, and several sets of boundary conditions. Validation is performed against the same configurations replicated in a FEM simulation.

Configuration	**Boundary set**	**x=**	**y=**	**FEM [m]**	**Present [m]**
LD_{110}/LD_{110}	CCCC ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-8.31 x 10^{-4}	-6.95 x 10^{-4}
		$-a/4$	$-b/4$	-8.31 x 10^{-4}	-7.04 x 10^{-4}
		$a/2$	$b/2$	-2.15 x 10^{-4}	-2.02 x 10^{-3}
	SCSC ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-1.37 x 10^{-3}	-1.20 x 10^{-3}
		$-a/4$	$-b/4$	-1.37 x 10^{-3}	-1.20 x 10^{-3}
		$a/2$	$b/2$	-3.05 x 10^{-3}	-2.87 x 10^{-3}
	FSFS ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-2.41 x 10^{-2}	-2.36 x 10^{-2}
		$-a/4$	$-b/4$	-2.41 x 10^{-2}	-2.37 x 10^{-2}
		$a/2$	$b/2$	-3.27 x 10^{-2}	-3.24 x 10^{-2}
$LD_{110}/ED_{222}/LD_{110}$	CCCC ($\lambda = 5 * 10^{10}$)	$a/4$	$b/4$	-1.06 x 10^{-4}	-9.62 x 10^{-5}
		$-a/4$	$-b/4$	-1.06 x 10^{-4}	-9.62 x 10^{-5}
		$a/2$	$b/2$	-2.60 x 10^{-4}	-2.55 x 10^{-4}
	SCSC ($\lambda = 3 * 10^{10}$)	$a/4$	$b/4$	-1.79 x 10^{-4}	-1.63 x 10^{-4}
		$-a/4$	$-b/4$	-1.79 x 10^{-4}	-1.48 x 10^{-4}
		$a/2$	$b/2$	-3.80 x 10^{-4}	-3.60 x 10^{-4}
	FSFS ($\lambda = 3 * 10^{10}$)	$a/4$	$b/4$	-2.50 x 10^{-3}	-2.40 x 10^{-3}
		$-a/4$	$-b/4$	-2.50 x 10^{-3}	-2.40 x 10^{-3}
		$a/2$	$b/2$	-3.40 x 10^{-3}	-3.40 x 10^{-3}

Table 14: Transverse displacements for two multiple sublaminate composite plates under different sets of boundary conditions. Validation performed with FEM simulation results. Transverse deflections: CCCC, SCSC and FSFS. Case setup from *Configuration 6*.

The agreement with reference results is not as good as expected. The energy method approximates the transverse displacements of the plate with relatively poor accuracy in both cases. Although the results are similar to reference ones, they are outside admissible tolerance (5%), specially for the first two points tested, and therefore cannot be considered a good source for validation. Some cases present a better agreement than others, in particular those with free edge conditions. This statement was mentioned in the previous case of study, and can be considered to hold better in these cases where a large number of kinematic variables are present during the assembly of the stiffness matrix. This difficulty in achieving a better accuracy comes from the large sensitivity of the penalty factor, for which small changes induce a large variation

in the solution field. The influence of the penalty factor is specially relevant within the energy method, and is covered in Section [5.5].

Second-order quantities are obtained during post-processing. The strain-displacement relationship is used to recover strains from the displacement field, which is obtained at the same time by replacing the network approximation with the resolved output weights. Stresses are given by the constitutive relation with the strain deformation field. A through-thickness representation of the transverse displacement and in-plane stresses is given in Fig.(23), evaluated at point $(a/2, b/2)$ for both configurations with FSFS boundaries. They are meant to illustrate the capabilities of the method in capturing the heterogeneity characteristic of multilayered plates, and specially the variations between sublaminates.

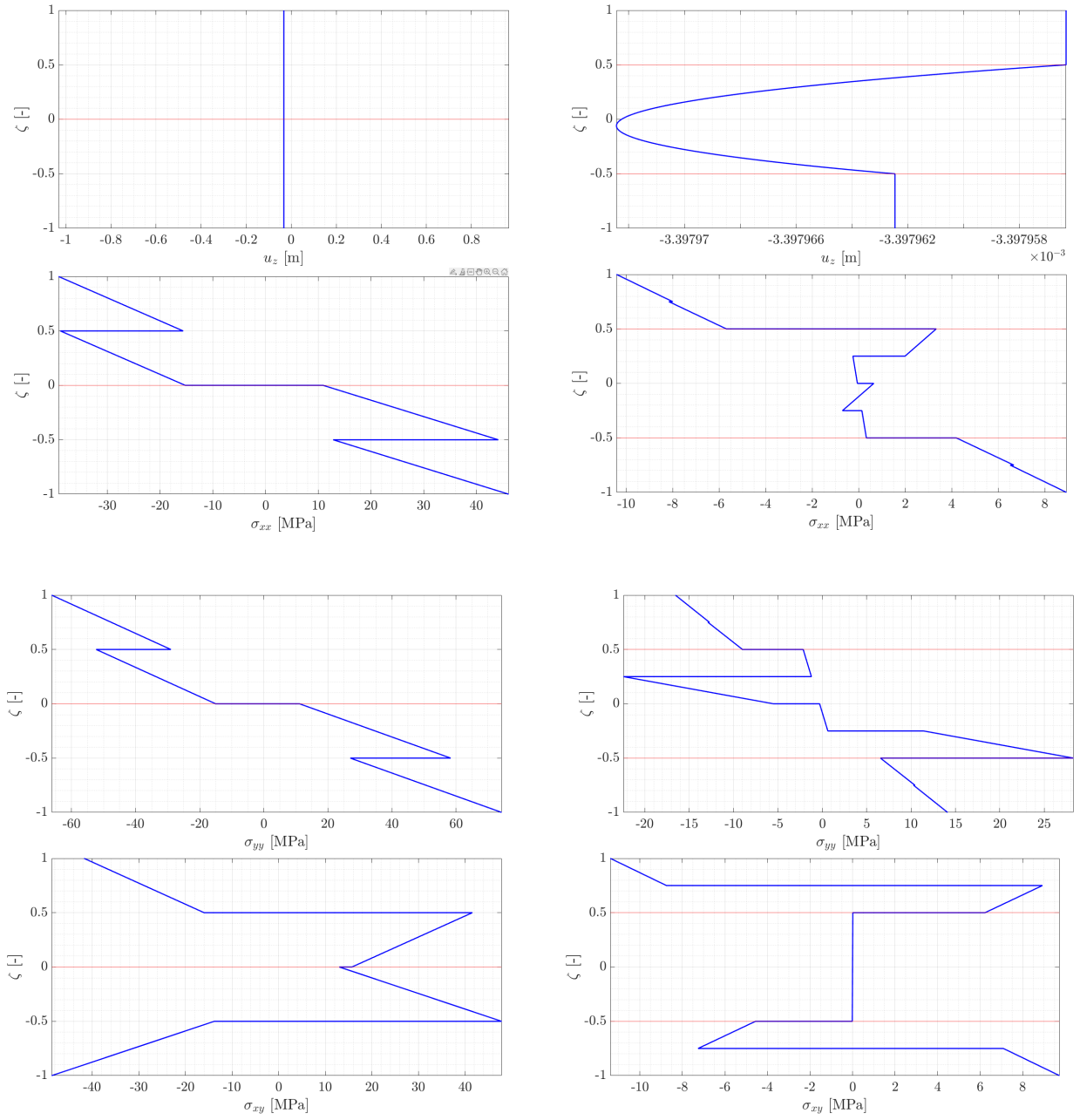


Figure 23: Through-thickness shapes of transverse displacement and in-plane stresses for both FSFS configurations reported in Table [14]. First case (*left*) and second case (*rigth*).

The piece-wise behaviour of the variables in the thickness direction is captured with the method. Non-dimensional coordinates (ζ) are used for the whole laminate, where the red lines differ-

entiate between sublaminates. Several conclusions can be drawn from this graph: the method is capable of well-representing the variation of stresses and transverse displacement according to different materials and layup combinations in the thickness direction, although their magnitude might be influenced by the penalty factor. Displacements satisfy the continuity condition introduced by compatibility between plies and sublaminates and their shape correspond to the use of different theories in each sublaminate. Therefore, the method is capable of dealing with combined ESL/LW theories. Finally, the major limitation is on the selection of right values of the case hyperparameters, that can lead to a prediction comparable with reference. Multiple sublaminate configurations cannot be considered completely validated, and the discussion is restricted to single sublaminate multilayered panels in the following.

Case study 8 - literature comparison with previous authors

A benchmark validation is done regarding cases available in literature. The motivation is to validate the method for cases that have been already covered by other authors, and check how well is the prediction of stresses and transverse displacement with the method at specific points. In those cases, different modeling approaches and solving procedures have been used. They are denoted by their authors, and the geometrical and material properties can be found in the proper reference. Those results are validated against a previous implementation with the Ritz-SGUF method available for these same cases, that is considered as a valuable reference. The initialization value for the penalty factor is given in each subcase for reproducibility. A small introduction is also given to each configuration:

- **Pagano**: a simply-supported (SSSS) very thick ($a/h = 4$) plate made of material (6) subjected to a uniform load of $1000N$ in magnitude directed in the negative Z-axis direction. It is described by four plies in cross-ply sequence $[0, 90, 0, 90]^\circ$ and by a fully-layerwise LD_{666} kinematic theory. Network setup: 50×50 sample points, 200 neurons and *tanh* activation. Ritz validation: 50×50 trial functions.

- **Iyengar**: several geometries of thick rectangular plates are considered by this author [67]. The one tested is made of alluminum (3) with dimensions $200 \times 1000 \times 10mm$ and fully simply-supported (SSSS). A kinematic theory ED_{332} is employed, while the external load is of uniform type and $1000N$ in magnitude directed in the negative Z-axis direction. Network setup: 100×100 sample points, 200 neurons and *tanh* activation. Ritz validation: 50×50 trial functions.

- **Kant**: a thin alluminum (3) square plate ($a/h = 100$) of dimensions $100 \times 100 \times 1mm$ is subjected to a uniform load of $1000N$ directed in the negative Z-axis direction. Two sets of boundary conditions are considered, namely: SSSS and CSCS. The plate is described with an ED_{332} kinematic theory in both cases. Network setup: 100×100 sample points, 200 neurons and *tanh* activation. Ritz validation: 50×50 trial functions.

Table [15] shows a comparison of these cases where the values of transverse displacement and in-plane stress components are reported at different locations in the plane and in the thickness direction.

Case	Value	x=	y=	z=	Ritz	Present
Pagano [66] ($\lambda = 5 * 10^{10}$)	u_z [m]	$a/2$	$b/2$	0	-3.63 x 10^{-8}	-3.66 x 10^{-8}
	σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	1680.7	1933.6
	σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	-14358.0	-9655.2
	τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	559.8	456.2
Iyengar [67] ($\lambda = 10^{10}$)	u_z [m]	$a/2$	$b/2$	0	-3.12 x 10^{-6}	-3.06 x 10^{-6}
	σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	0.299	0.359
	σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	-0.091	-0.150
	τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	7425.7	8700.1
Kant (SSSS) [68]($\lambda = 6 * 10^{9}$)	u_z [m]	$a/2$	$b/2$	0	-6.08 x 10^{-5}	-5.99 x 10^{-5}
	σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	2.873	3.859
	σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	-2.873	-3.847
	τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	-0.801	-0.790
Kant (CSCS) [68] ($\lambda = 3 * 10^{9}$)	u_z [m]	$a/2$	$b/2$	0	-2.86 x 10^{-5}	-2.87 x 10^{-5}
	σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	1.973	2.682
	σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	-1.447	-2.042
	τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	-0.504	-0.522

Table 15: Benchmark with literature cases. Transverse displacement and in-plane stress components at several test locations. Ritz validation: 50×50 trial functions.

Displacements are generally well approximated by the energy method. It is at stress level where the largest differences are observed. Stresses τ_{xy} generally show a minor deviation with respect to reference values, while principal stresses are well above admissible tolerance (5%) for this test case. This is a phenomenon that appears whenever stresses are recovered, and affects almost any problem that uses the energy method. The shape of the stress field is generally well-approximated, and *Case study 8* showed how the program is capable of recovering their evolution in the thickness direction, but not their magnitude. The problem remains opened at this point. The use of different penalty factors has not shown an improvement in this behaviour. Therefore, the recovery of stresses must be handled carefully if needed, and further investigation is required to understand the origin of these issues.

Case study 9 - shell with different geometries

The applicability of a PINN-ELM framework to the analysis of *shells* is of great importance due to their major presence in aerospace structures. A particular formulation in the context of SGUF was introduced for these geometries in Section [3.5.2], regarding the energy method and has been the only development of a specific formulation in this matter. The numerical problems reported with the strong collocation method motivated it was not extended to shells prior to be resolved.

The case of a fully clamped shell using *Configuration 7* is reported in Table [16]. Several cases of study report the value of the transverse displacement at the center of the shell $(a/2, b/2)$ with different combinations of geometrical parameters, involving the modification of radius $R = 100, 500mm$ and subtended angle $\phi = \pi, \pi/2, \pi/3, \pi/6$, that correspond to different values of the characteristic dimension b (recalling that $b = R\phi$, as of the definition for cylindrical shells). Shell thickness is the third parameter, for which three configurations are explored: $h = 1, 5, 10mm$. A single value of the penalty factor is used for reference in each subcase, which is applied to all combinations of the geometrical parameters inside.

Radius of curvature [R]	Subtended angle [ϕ]	Case	$h = 1mm$	$h = 5mm$	$h = 10mm$
100mm ($\lambda = 5 * 10^8$)	π	Present [m]	-5.84 x 10^{-8}	-3.29 x 10^{-8}	-2.27 x 10^{-8}
		Ritz [m]	-2.57 x 10^{-7}	-3.45 x 10^{-8}	-2.17 x 10^{-8}
	$\pi/2$	Present [m]	-6.35 x 10^{-8}	-8.14 x 10^{-8}	-4.18 x 10^{-8}
		Ritz [m]	-1.53 x 10^{-7}	-7.79 x 10^{-8}	-4.03 x 10^{-8}
	$\pi/3$	Present [m]	-3.04 x 10^{-7}	-8.20 x 10^{-8}	-3.66 x 10^{-8}
		Ritz [m]	-3.92 x 10^{-7}	-7.94 x 10^{-8}	-3.51 x 10^{-8}
	$\pi/6$	Present [m]	-3.55 x 10^{-7}	-4.61 x 10^{-8}	-1.29 x 10^{-8}
		Ritz [m]	-3.99 x 10^{-7}	-4.44 x 10^{-8}	-1.20 x 10^{-8}
500mm ($\lambda = 3 * 10^8$)	π	Present [m]	-4.65 x 10^{-6}	-1.25 x 10^{-6}	-6.33 x 10^{-7}
		Ritz [m]	-5.88 x 10^{-6}	-1.39 x 10^{-6}	-6.95 x 10^{-7}
	$\pi/2$	Present [m]	-2.86 x 10^{-6}	-1.04 x 10^{-6}	-6.01 x 10^{-7}
		Ritz [m]	-5.88 x 10^{-6}	-1.40 x 10^{-6}	-6.88 x 10^{-7}
	$\pi/3$	Present [m]	-2.16 x 10^{-6}	-1.10 x 10^{-6}	-7.08 x 10^{-7}
		Ritz [m]	-6.13 x 10^{-6}	-1.31 x 10^{-6}	-7.20 x 10^{-7}
	$\pi/6$	Present [m]	-3.51 x 10^{-6}	-1.89 x 10^{-6}	-8.08 x 10^{-8}
		Ritz [m]	-4.61 x 10^{-6}	-1.89 x 10^{-6}	-8.05 x 10^{-8}

Table 16: Fully clamped angle ply *graphite epoxy* case comparison for shells. Modified thickness, radius of curvature and subtended angle. Transverse displacements evaluated at $(a/2, b/2)$ on the midsurface. Case setup from *Configuration 7*.

By looking at the comparison with Ritz reference values, several conclusions can be extracted. Large radius with small angles correspond to portions of the arc that approximate the shape of a plate. Composite layup cases have already been tested for panels with no curvature, and a good agreement with the reference was obtained, therefore a similar behaviour is to be expected. Only in (some) $\pi/6$ cases the agreement is within admissible tolerance of 5% difference, and also for thicker configurations ($h = 5mm$, $h = 10mm$). The method fails to predict the central displacement for a thin shell and large values of the subtended angle.

The large error associated to thin configurations is not only reported by point-wise evaluations but a contour of transverse displacements is given in Fig.(24). The case $R = 100mm$, $\phi = \pi/2$ is taken for reference, and the contours for its three different thickness subcases are compared with those obtained with the Ritz method.

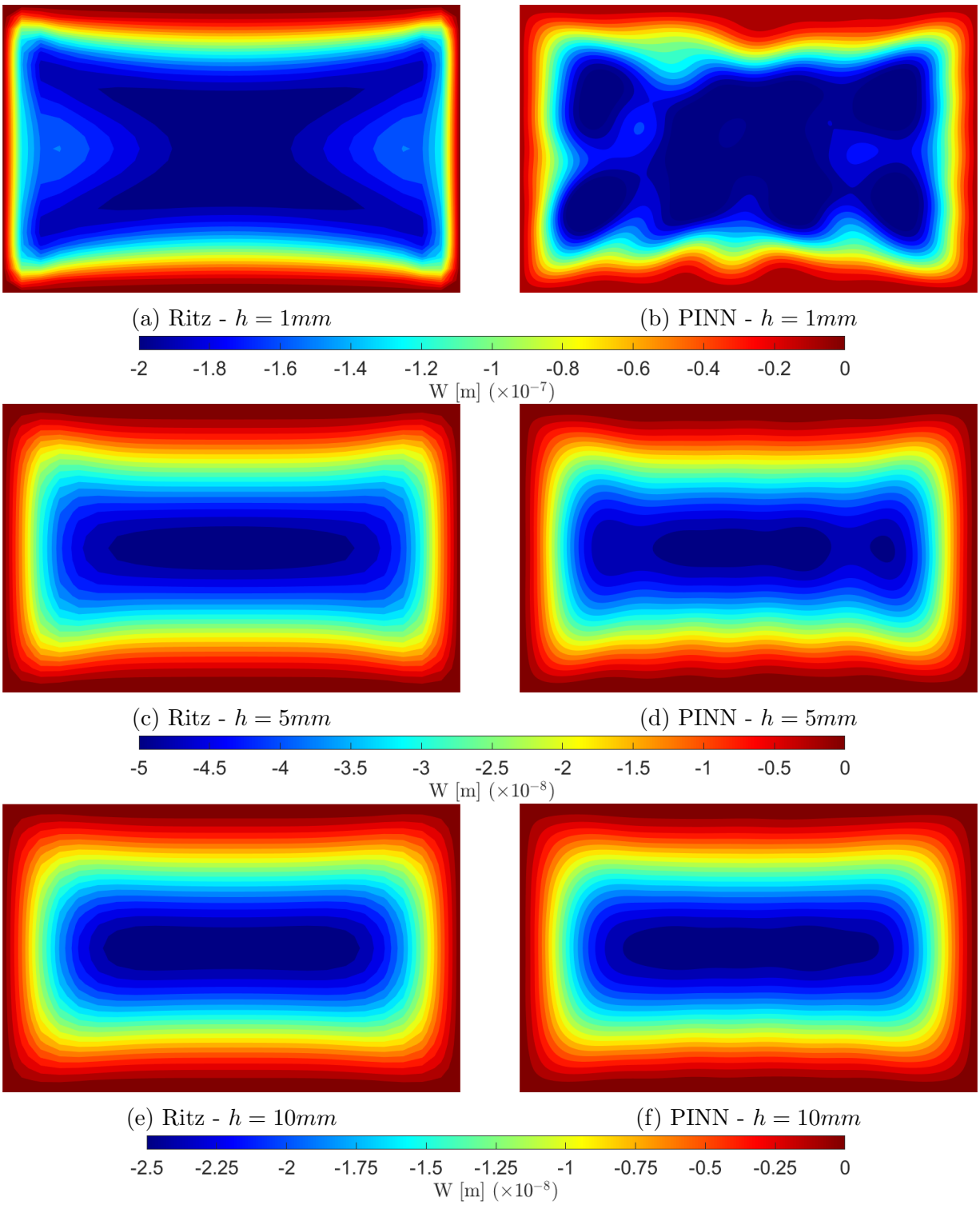


Figure 24: Contours of transverse displacement of shell configuration: $R = 100mm$, $\phi = \pi/2$ from Table [16]. Shell thickness: $h = 1mm$ (*top*), $h = 5mm$ (*middle*) and $h = 10mm$ (*bottom*). Case setup from *Configuration 7*.

The energy method is incapable of capturing the referenced contour in the first configuration (thin). The thicker the shell becomes, the better it can predict transverse displacements. These results are extensible to other cases where thin shells are considered and CCCC boundaries are imposed. No improvement has been observed with the use of higher order theories, neither shifting to an ESL model. This numerical issue makes a bottleneck on the applicability of the

energy method for shells. Other sets of boundary conditions have been tested, and their contours can be found in Appendix B. The effect disappears when free boundaries are introduced, and even the agreement with the reference is exceptional when opposite edges are free (CFCF and FCFC). This finding hinders the applicability of the energy method to deal with more complex shell properties, and the analysis is restricted to other configurations that avoid that of a fully-clamped shell.

Case study 10 - skew plate with different aspect ratios

A major contribution by this method has to do with the capabilities of the program to deal with any geometry. The collocation method can be extended to any shape as long as a proper sampling is ensured. This is specially critical at the boundaries, that may not follow the directions of the reference axes, or have a curved shape. A sampling at regular intervals is used to map the boundaries, although it introduces an error in terms of the associated surrounding area to the numerical integration point, that partially overlaps with collocation inner points. A diagram of the issue is reported in Fig.(25), where the original sampling strategy is compared with an improved one:

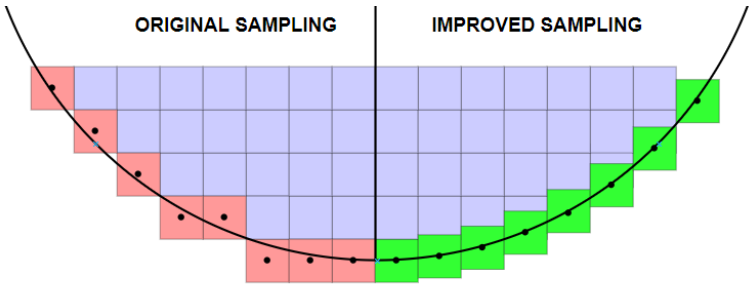


Figure 25: Sampling strategies for non-regular edge geometries. Original sampling strategy at the boundaries (*red*) against improved sampling strategy (*green*).

The energy method always introduces a certain error from the boundary points into the numerical integration, which is furtherly reduced with a larger sampling grid on the whole plate. If a large amount of points is used to map the boundaries alone, the error becomes larger, and the same occurs if too few points are considered. With the *improved sampling strategy*, the uniformity of the grid is broken in order to place a set of points that can map that boundary, even though a certain overlapping with inner point regions (blue squares) is expected. The criterion followed consists on using the distance between points on the inner part of the collocation grid and map the boundaries with this interval (including corners) in a non-uniform way. This strategy has proven effective and is used whenever the geometry differs from that of a rectangular panel.

The skew plate of *Configuration 8* is placed under study. Two configurations of the aspect ratio and the skew angle are considered, while the rest of the comparison with reference results by Kale [1] can be found in Appendix B. A diagram on the dimensions of a skew plate is reported in Fig.(26). This makes a good example to test the robustness of the method when skewness is taken to the limit.

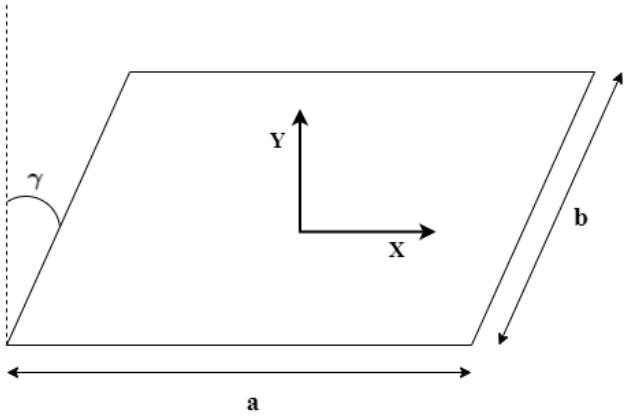


Figure 26: Diagram of the principal dimensions involved with the skew plate geometry in [1].

Table [17] reports the transverse displacement and in-plane stress components evaluated at several test locations both in-plane and in the thickness direction. FEM results are used for validation, evaluated at the same locations.

Aspect ratio (AR)	**Skew angle (γ)**	**Value**	**x=**	**y=**	**z=**	**Present**	**FEM**
1	15° ($\lambda = 3 * 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	1.68 x 10^{-4}	1.68 x 10^{-4}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.225	-0.127
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.234	0.131
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0518	0.0442
	75° ($\lambda = 8 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	1.41 x 10^{-6}	1.39 x 10^{-6}
		σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	-8437.9	-6602.3
		σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	18338	14350
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	751.43	710.9
2	15° ($\lambda = 2 * 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	2.16 x 10^{-5}	2.07 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0591	-0.0217
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0963	0.0577
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	12498	10540
	75° ($\lambda = 9 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	1.12 x 10^{-7}	1.09 x 10^{-7}
		σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	-2238	-1248.6
		σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	5225	4162.8
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	211.84	347.64

Table 17: Displacement and stress results comparison for the cases of skew plates from [1]. Case setup from *Configuration 8*.

The predicted values of the displacements are in good agreement with reference results by FEM, with a minimal difference within tolerance at the point of maximum deflection. Recovered stresses at different locations suffer of the same issues reported in *Case study 9*, a large error is introduced in their computation by an unidentified source, generally overpredicting their values. The shear stress is also outside tolerance, but typically reports a better agreement with reference data. The motivation for this error is unknown, but whenever the analysis is restricted to the displacement field, the method has proven reliable for skew plates configurations, both in small and large values of the skew angle.

Convergence study - free vibrations analysis

In the same line of the analysis done with the strong formulation method, a convergence study of the natural frequencies from low to high order theories is performed. The plate from

Configuration 4 is used to carry out a convergence study with seven kinematic theories and for different sets of boundary conditions, which are reported in Fig.(27). These results use no preconditioner, and the range of neural network weights generation has been modified to $w = [-3, 3]$, the motivation for this different setup is explained in the next case of study.

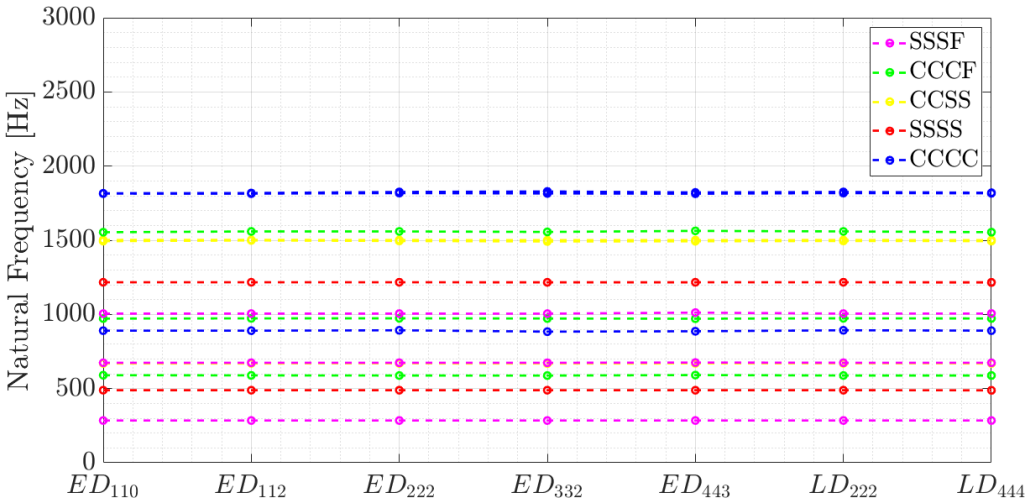


Figure 27: Convergence of the weak method in free vibrations with respect to kinematic theory. Boundary conditions: CCCC, SSSS, CCSS, CCCF and SSSF. Case setup from *Configuration 4*.

The first three frequencies converge in all cases early from the first theory, similar to what was shown in the linear static convergence study. No differences are observed depending on the boundary conditions. Those same results are validated with a Ritz simulation that uses 50×50 trial functions, and the first ten natural frequencies are compared in Table [18] for kinematic theory ED_{332}, which is taken as reference in order to provide results for a high order theory.

BC	**Case**	**Mode [Hz]**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CCCC	Present	883.86	1815.40	1828.98	2675.97	3242.78	3246.42	4060.61	4075.94	5202.27	5250.01
($\lambda = 10^{10}$)	Ritz	881.02	1793.86	1794.89	2642.60	3211.81	3227.44	4024.89	4024.56	5130.50	5130.99
SSSS	Present	488.92	1216.01	1216.49	1951.72	2421.50	2422.79	3165.19	3165.80	4097.04	4107.11
($\lambda = 10^{10}$)	Ritz	483.01	1207.67	1207.79	1931.03	2414.15	2414.32	3134.69	3134.80	4096.37	4096.38
CCSS	Present	668.82	1491.56	1500.94	2290.84	2819.38	2821.60	3588.36	3603.53	4886.07	4943.27
($\lambda = 10^{10}$)	Ritz	662.25	1480.67	1486.34	2268.55	2799.12	2803.201	3558.52	3565.77	4597.90	4600.21
CCCF	Present	587.65	972.32	1556.16	1856.04	1970.12	2837.16	3009.21	3241.24	3444.80	4176.43
($\lambda = 10^{10}$)	Ritz	582.25	973.65	1540.93	1870.23	1960.78	2836.71	2977.02	3278.17	3412.30	4202.55
SSSF	Present	284.73	673.81	1005.15	1441.29	1493.64	2202.69	2296.35	2659.04	2783.03	3550.75
($\lambda = 10^{10}$)	Ritz	283.69	674.78	1004.03	1437.83	1508.24	2200.51	2299.14	2652.38	2823.34	3542.69

Table 18: First ten natural frequencies of a square plate with ED_{332} theory under different boundary conditions. Case setup from *Configuration 4*. BC: boundary conditions.

The differences are within admissible tolerance (5%) in all frequency pairs, except those in higher frequencies for the CCSS case, which show a larger deviation. Several parameters can be influencing the case, specially the penalty factor that has been remained constant for the whole analysis in order to compare all cases with the same configuration. The results show that high order theories can be used in free vibrations with the energy method, and expect a good agreement with reference values. No particular issues are reported regarding the use of different boundary conditions that may hinder their applicability to other cases of study.

Case study 11 - multilayered composite plate in free vibrations

The analysis is extended to the test case of a multilayered composite plate reported in *Configuration 5*, subjected to free vibrations analysis. Different sets of boundary conditions are considered. The initialization values of the penalty factor are given in the corresponding case, and all solutions use a modified range of weights of $w = [-3, 3]$ according to the findings of the previous case. The motivation of this example is to validate the eigensolver capabilities with different layup sequences. Due to the expected piece-wise behaviour of the laminate, theory LD_{110} is chosen for the simulation, and its results are validated against a FEM solution. The first three natural frequencies are used in each case and given in Table [19]:

Layup	Boundary conditions	Mode	FEM [Hz]	Present [Hz]
$[0, 90, 0]^\circ$ (4) ($\lambda = 10^{10}$)	CCSS	1	29.99	29.85
		2	44.31	44.56
		3	73.04	74.18
	FFCC	1	7.28	7.16
		2	16.97	16.85
		3	39.01	38.41
	SSSF	1	7.26	7.66
		2	24.87	23.61
		3	30.59	30.74
$[-45, 0, 45]^\circ$ (5) ($\lambda = 8 * 10^9$)	CCSS	1	23.55	23.80
		2	48.10	48.62
		3	50.87	51.43
	FFCC	1	5.92	5.89
		2	19.34	19.29
		3	21.16	21.09
	SSSF	1	9.27	10.64
		2	24.57	25.01
		3	29.70	33.91

Table 19: Layup sequences and composite materials for different boundary conditions in free vibrations. First three natural frequencies, boundary conditions: CCSS, FFCC and SSSF. FEM modal simulation used in validation. Case setup from *Configuration 5*.

The agreement in validation is within tolerance and no major differences are reported in either case. It confirms the good applicability of the energy method in free vibrations analysis in the context of composites, although the case has been restricted to a single sublaminate. The assembly of several sublaminates is not considered, in the same way the linear static analysis results of *Case study 8* showed an off-limits difference of transverse displacements. Not reported free vibrations simulations have proven this behaviour, and the analysis remains restricted to the case of a single sublaminate.

Case study 12 - elliptic plate with different aspect ratios

The capabilities to represent curved-edge panels are tested in the case of elliptic plates made of a composite material, where different aspect ratios of these geometries are analysed. The details of the plate and network configuration are reported in *Configuration 9*. The plates have a graphite epoxy (4) cross-ply configuration of two plies. Three cases are considered: a circular plate ($a/b = 1$), elliptic plate ($a/b = 2$) and slender elliptic plate ($a/b = 4$) and validated against a FEM modal simulation. This analysis is motivated by the testing of the degradation in the prediction of natural frequencies when the edges are curved and their aspect ratio vary. An increasing error associated to the difficulty in sampling the boundaries is expected, while

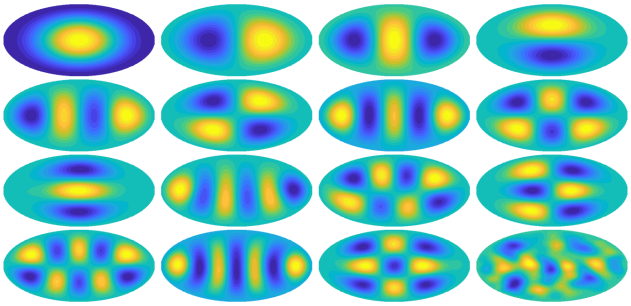
the *improved sampling* strategy reported in *Case study 11* is used in all cases.

Table [20] reports the first six natural frequencies of the cases under consideration, both from the energy method and a FEM modal simulation.

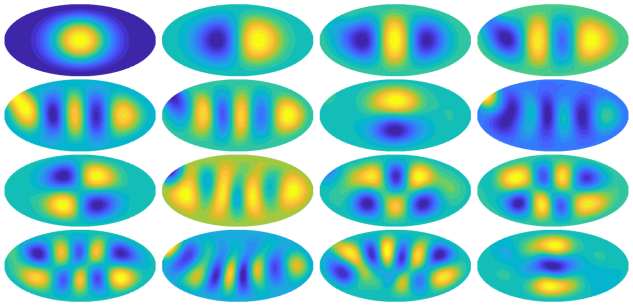
Case	**Mode**	**PINN-ELM [Hz]**	**FEM [Hz]**	**Diff.** [%]
Circle ($R = 250mm$) ($\lambda = 10^9$)	1	100.34	100.09	0.25
	2	208.91	208.26	0.31
	3	209.06	208.26	0.38
	4	342.28	341.56	0.21
	5	342.50	341.56	0.28
	6	390.58	389.49	0.28
Ellipse ($a = 250mm/b = 125mm$) ($\lambda = 9 * 10^8$)	1	265.17	268.12	-1.10
	2	384.25	386.71	-0.64
	3	544.67	547.90	-0.59
	4	676.13	683.75	-1.11
	5	752.83	753.43	-0.08
	6	852.56	861.43	-1.03
Ellipse ($a = 250mm/b = 62.5mm$) ($\lambda = 7 * 10^8$)	1	975.73	969.00	0.69
	2	1160.50	1145.30	1.33
	3	1371.80	1357.00	1.09
	4	1595.60	1605.40	-0.61
	5	1824.80	1892.10	-3.56
	6	2153.00	2218.90	-2.97

Table 20: Free vibrations analysis of elliptic plates with different aspect ratios. Configurations: $a/b = 1$, $a/b = 2$ and $a/b = 4$. FEM modal simulation used in validation. Case setup from *Configuration 9*.

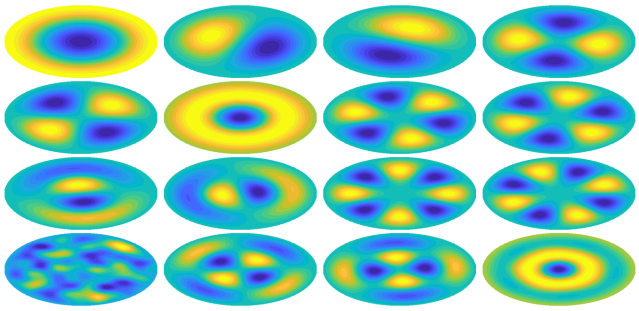
The agreement on the natural frequencies indicates the energy method is capable of dealing with high aspect ratios in panels with curved boundaries. Minor differences are observed between the first two and the $a/b = 4$ ellipse, regarding the differences column of the table. The sign of the difference alternates between first and last three frequencies in this case, and the error starts to grow. This effect is better observed by looking at the transverse displacement contours of Fig.(28), where a non-simmetry is present in the half-wave shapes of modes 4,5,6.



(a) Elliptical ($a/b = 2$) plate



(b) Elliptical ($a/b = 4$) plate



(c) Circular plate

Figure 28: Modal shapes of the different cases available in Table [20].

This problem is introduced by an improper sampling of the boundary, although an improved strategy has been used, the error introduced in the numerical integration is too large and leads to a bad representation of the modes. It has its onset for $a/b = 4$ and it can be expected to occur for highly elliptic configurations, while it is not present in previous cases. This consideration must be taken into account whenever elliptic plates are studied, and it adds an additional layer of complexity into the problem.

Case study 13 - holed elliptic plate with different aspect ratios of the internal cutout

An additional validation has been performed considering the case of an elliptic plate reported by Fantuzzi [2] with a drilled section in the middle, of different dimensions. Holed structures are the last step in the validation of the method to work with arbitrary shapes. The plates are made of an isotropic material (2) and have dimensions $4000 \times 2000 \times 5mm$ (major axis: 2000mm, minor axis: 1000mm), with an internal cutout given by 4 configurations: $b_{internal}/b = 0$, $b_{internal}/b = 0.2$, $b_{internal}/b = 0.4$, $b_{internal}/b = 0.6$. Internal and external boundaries are fully-clamped and a high order theory ED_{332} is used to simulate the structure, that may present complicated deflection shapes. Penalty factor is kept constant to $\lambda = 5 * 10^9$ for all simulations, while FEM results and exact ones given in the reference are used for validation. *Note: material properties are given in non-dimensional form in the reference, those have been converted to the proper scale.*

Case	Boundary conditions	Mode	Present [Hz]	FEM [Hz]	Reference (IGA) [2]
Ellipse ($\frac{b_{internal}}{b} = 0$) ($\lambda = 5 * 10^9$)	C	1	12.87	12.87	12.78
		2	17.84	17.72	17.75
		3	25.51	24.94	24.93
		4	32.94	33.18	33.19
		5	34.98	34.60	34.56
		6	39.92	39.78	39.86
Ellipse ($\frac{b_{internal}}{b} = 0.2$) ($\lambda = 5 * 10^9$)	C/C	1	19.47	19.89	19.89
		2	19.61	20.20	20.09
		3	35.56	32.60	32.74
		4	35.82	33.03	33.11
		5	37.14	38.56	38.33
		6	39.32	38.88	38.77
Ellipse ($\frac{b_{internal}}{b} = 0.4$) ($\lambda = 4 * 10^9$)	C/C	1	27.56	30.19	30.16
		2	27.78	30.28	30.34
		3	46.69	45.18	45.19
		4	46.85	45.37	45.29
		5	58.26	58.87	58.88
		6	58.46	58.99	59.03
Ellipse ($\frac{b_{internal}}{b} = 0.6$) ($\lambda = 2 * 10^9$)	C/C	1	52.59	58.58	58.66
		2	54.26	58.58	58.57
		3	80.64	77.50	77.29
		4	81.50	77.53	77.70
		5	105.95	104.84	104.77
		6	106.56	104.90	104.84

Table 21: Elliptic shaped plates with internal apertures of varying dimension: $\frac{b_{internal}}{b} = 0$, $\frac{b_{internal}}{b} = 0.2$, $\frac{b_{internal}}{b} = 0.4$ and $\frac{b_{internal}}{b} = 0.6$.

The match of the frequencies in the first case agrees with the conclusion of the previous example, the energy collocation method is capable of approximating the natural frequencies of an elliptic plate with high accuracy. The agreement worsens for larger sizes of the inner hole, and specially the first two frequencies get out of tolerance in the last two cases. Those modes correspond to large half-wavelengths, and the method finds an increasing difficulty to predict their frequency, which can be associated to the presence of the hole, that is interpreted internally as a discontinuity. Nevertheless, other parameters such as the penalty factor or the boundary sampling may have an intrinsic influence on the prediction that has not been considered. The second case demonstrate how the method can be applied satisfactorily to deal with the presence of holes, and its capabilities are not restricted only to fully clamped, but the same kinds of boundary conditions can be imposed to the inner perimeter and expect a similar performance. A contour plot of the transverse deflection for these drilled elliptic plates can be found in Appendix B.

5.4 Strong-form collocation method - numerical assestment

Test case 1 - comparison of different grid distributions

Certain problems have been mentioned to affect particularly the first collocation method. First and foremost, due to its importance on the simulation of any analysis, the distribution chosen for the collocation grid points has a major impact on the quality of the solution, and its selection must be justified to work in every case. Different simulations are considered, where the square plate in *Configuration 1* is used, and the grid distribution is the free parameter for which the four strategies first presented in Section [4.3] are employed. The first case considers the plate with CCCC boundaries, it avoids at first instance the definition of free boundaries to dismiss any coupled effects that could appear. A contour of transverse displacements of the

four cases are shown in Fig.(29):

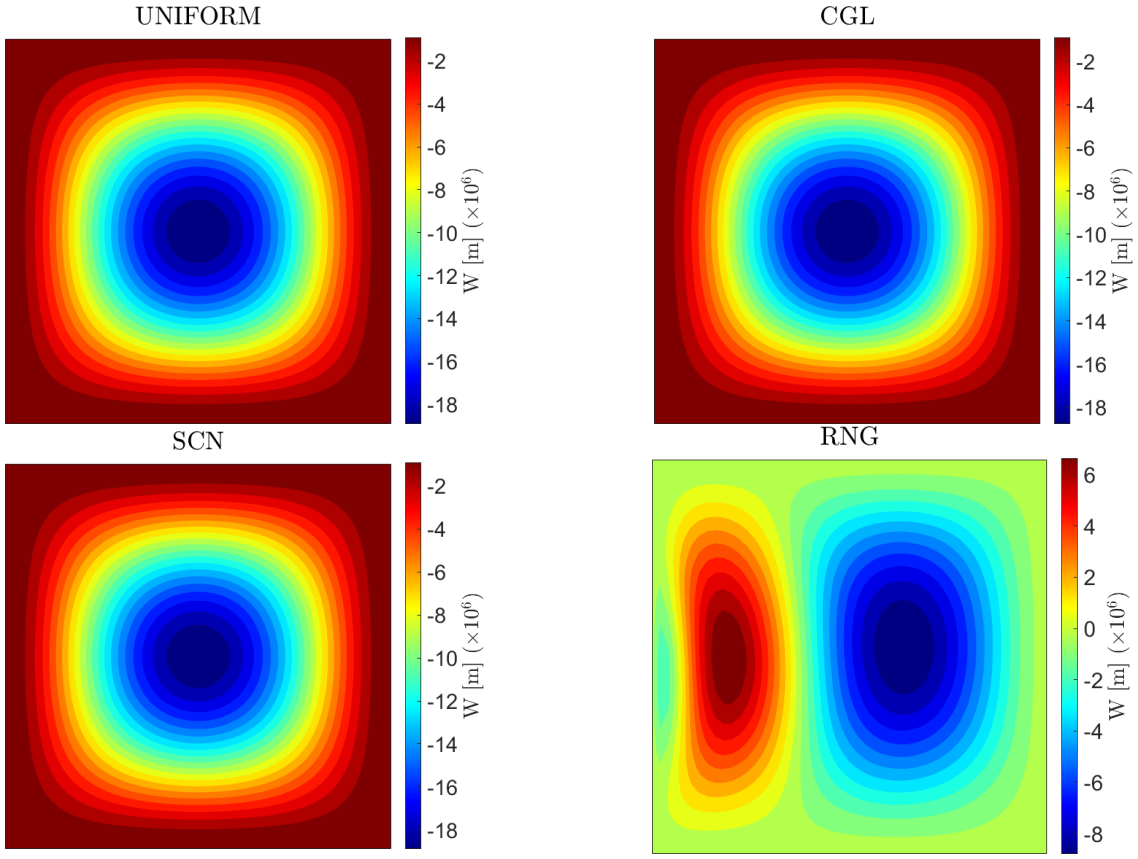


Figure 29: Contours of transverse displacement of plate in *Configuration 1*, with different grid distribution strategies. Sampling techniques: uniform, CGL, SCN and RNG. Boundary conditions: CCCC. Linear static analysis.

The randomly generated distribution has obtained the worst results compared to the others. This strategy was originally developed to deal with the free edge issues reported in this same chapter, but its poor mapping of certain areas (specially at the boundaries) make it unreliable, even if a fine grid (50×50 samples) was to be considered, it is discarded for further analysis. On the other hand, the remaining strategies show a similar performance, they coincide in the value of maximum displacement with reported references in Section[5.2], any of them could be used a priori within the first collocation method.

This numerical assestment enters into a new level of complexity when free edges are considered. The program has shown an inherent difficulty in approximating the displacement solution field when one or more edges are constraint. The study on collocation distributions can be extended to prove if any of the descriptions show a better behaviour under these circumstances. In Fig.(30) the contours of transverse displacement are given for the same plate, subjected this time to SSSF boundary conditions. The size of the grid has been reduced to 15×15 points.

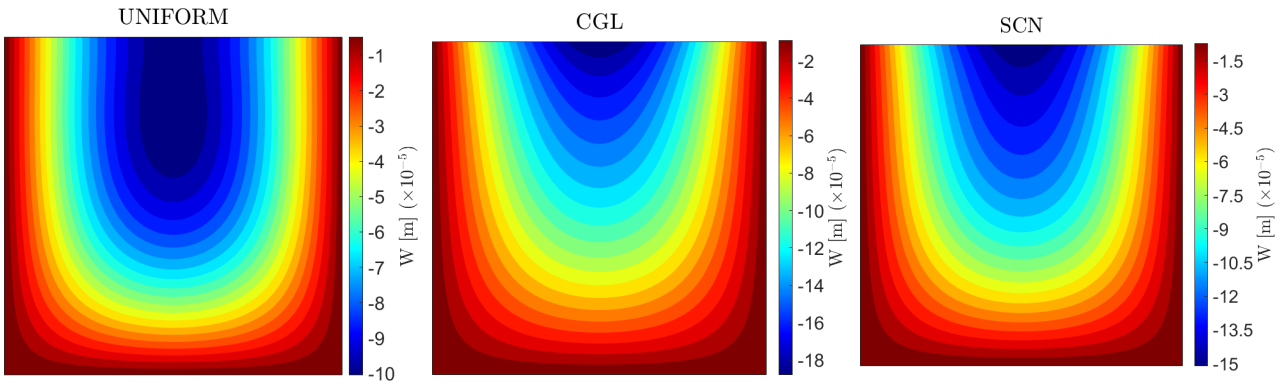


Figure 30: Contours of transverse displacement of plate in *Configuration 1*, with different grid distribution strategies. Boundary conditions: SSSF. Sampling techniques: uniform, CGL and SCN. Linear static analysis.

By using this grid size, major differences are reported between strategies. CGL better approximates the real deflection of the plate, and the maximum point is on the free edge (top), its value underestimates the reference (1.97×10^{-4}m) given on the previous chapter, but it is well within admissible tolerance (5%). The difference grows with SCN strategy, that predicts an even lower displacement, and uniform sampling shifts the region of maximum displacements towards the center of the plate. This behaviour is not well understood, every strategy is expected to correctly represent the displacements as long as the governing equations are satisfied by the solution at every point. With regard to the different strategies, the same performance is obtained for any other combination of boundaries that include free edges (i.e. CCCF,SFSF,CFCF,CCFF...). The available metrics in the simulation (residuals and RMSE) have proven unuseful to analyse these issues, large values of the loss are obtained and the row-by-row residuals give a large difference with external outputs, even for those cases that match almost exactly reference results.

Test case 2 - influence of the grid size

Although the behaviour of CGL can be considered to meet the expectancies, the prediction changes for small variations in the grid size. In Fig.(31), the case of CGL from Fig.(30) is repeated with a grid progressively refined from $N = 10 \times 10$ until $N = 20 \times 20$, where the behaviour becomes asymptotic, reason why finer grids are not considered in this comparison.

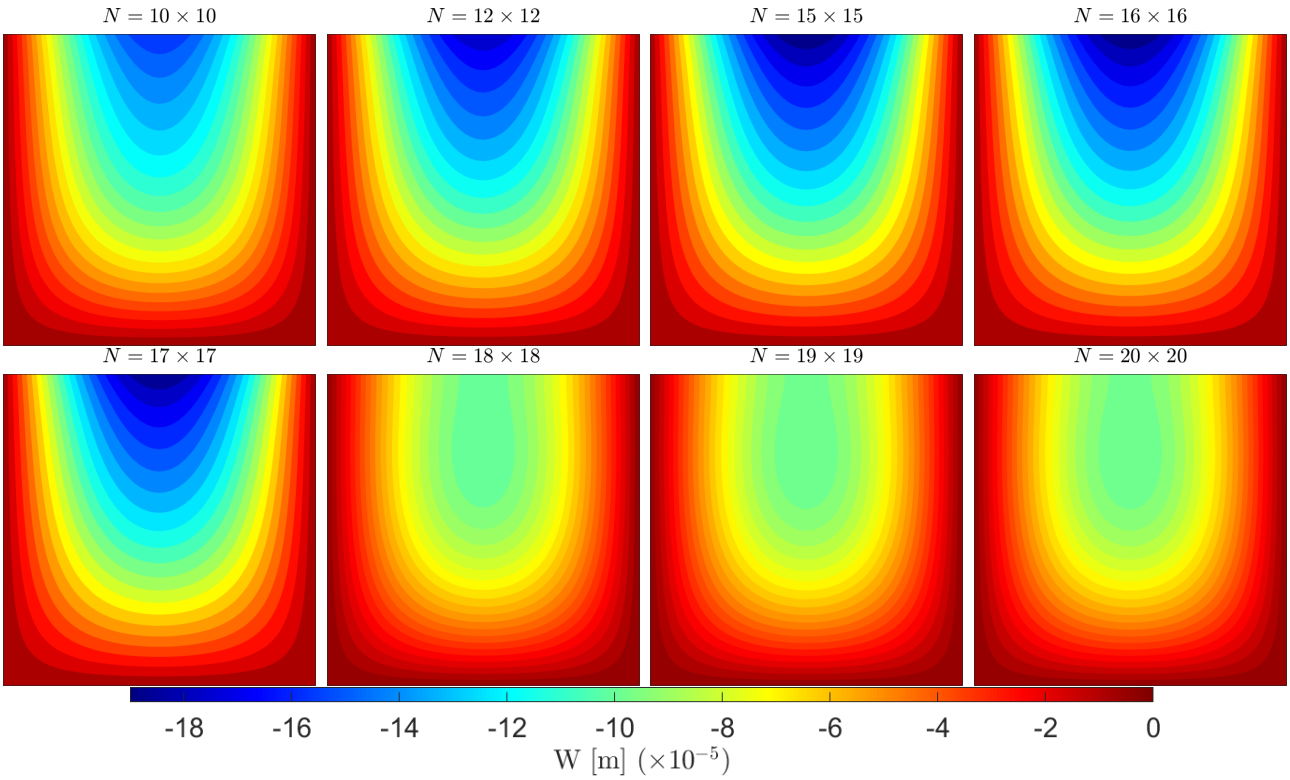


Figure 31: Contours of transverse displacement of plate in *Configuration 1* with CGL grid strategy. Grid sizes: 10×10, 12×12, 15×15, 16×16, 17×17, 18×18, 19×19 and 20×20. Boundary conditions: SSSF. Linear static analysis.

The contour for $N = 18 \times 18$ is the onset point at which the prediction of the displacement field worsens considerably with respect to reference. Early from $N = 10 \times 10$ points, the solution is underestimated due to a lack of grid points to properly represent the transverse behaviour, this statement can be extended to most cases simulated with this collocation method (it is referred as an *insufficient sampling*). $N = 15 \times 15$ has been said to properly capture displacements and the tendency maintains in finer grids, but a discontinuity occurs at 18 sampling points per direction and the plate is restricted in bending near the free edge, the solution field is largely underestimated between both configurations. It is an interesting observation that introduces the possibility that the way of selecting points in the geometry cannot deal with the satisfaction of equilibrium equations if those are taken at certain locations, that remain unknown. The selection of CGL strategy in *Configurations 1,2,3* has been done regarding these considerations, it is along with SCN the one with a better performance, therefore it is taken as default to keep this parameter fixed in the rest of implementations.

Test case 3 - influence of the number of neurons in linear static analysis

Another parameter with a major influence on the solution belongs to the family of *hyperparameters* of the neural network. The size of the collocation grid has already been mentioned, it is also included in this family due to its coupled effect with other parameters over the solution, it is referred as such in the following discussion. The *number of hidden nodes* (neurons) used in the network architecture dictates the number of connections with the output layer, and gives an intuition on the complexity of the features the network is capable of representing. Plate in *Configuration 1* (grid size $N = 20 \times 20$, restored to default) is used once again to compare the evolution of the transverse displacement field with an increasing number of neurons. Boundary conditions are CCCF and steps of 100 neurons augment the network, contours of transverse

displacement are given in Fig.(32). This particular case was included in the convergence study of Section [5.2], showing a large difference with reference results due to the presence of the free boundary.

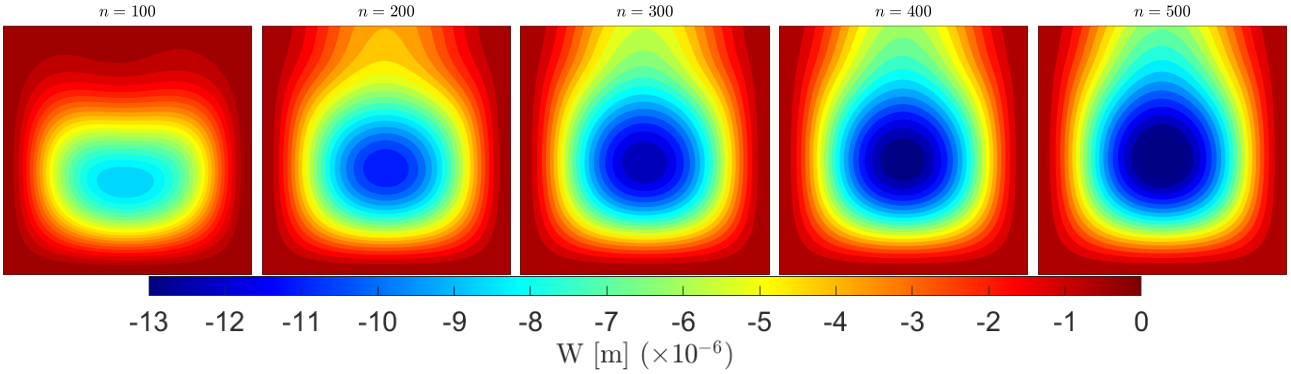


Figure 32: Contours of transverse displacement of plate in *Configuration 1* with increasing number of hidden nodes. Neurons: $n = 100$, $n = 200$, $n = 300$, $n = 400$ and $n = 500$. Boundary conditions: CCCF. Linear static analysis.

The first situation is a clear example of a bad performing network, the contour is not only underestimating the displacements near the free edge, but a non-symmetric plot indicates the number of neurons is too small to represent the solution field accurately. Additional neurons relax this statement, and the maximum deflection value increases. $n = 500$ neurons is the last solution reported, after which the evolution of the displacements remains (almost) the same. This asymptotic behaviour is similar to what occurs with the collocation grid, it appears the network is unable to predict how the plate behaves, and it largely underestimates the displacements (reference at maximum: $4.50 \times 10^{-5} m$). This effect is only particular of free edges, and 500 neurons generally well represents the displacement field in any other situation. That value has been taken as a default parameter in *Configurations 1,2* attending to these motivations.

Test case 4 - influence of different hyperparameters

Not only the number of hidden nodes has been considered a parameter worth studying in dealing with free boundary issues. The selection of the activation function at the hidden layer is of paramount importance depending on the problem at hand. Unfortunately, very few literature is available regarding this topic in ELM, although two principal functions appear repeatedly: *tanh* and *sigmoid*. *Tanh* has been said to be widely used in regression problems (and also in PINN) regarding traditional networks, while *sigmoid* is a basic choice in logistic regression (classification) problems. None of these situations matches the case at hand, that was mentioned in Section [4] to somewhat fall inside the category of Reinforcement Learning. Five activation functions were introduced in Section [4.3] and are included in the program, but only these two are considered in the analysis, the performance of the other functions can be considered as intermediate cases between them. Additionally, the modification of the range for random generation of weights is included, following the experience of the energy method. This modification seeks to have the same effect as in the context of free vibrations, making the solution more robust to numerical issues. Table [22] reports the output of this analysis for a plate in *Configuration 1* and CFCF boundary conditions with both activations, three different ranges of weights, and three different ranges of bias, that have been added for completeness. Difference with Ritz reference for the maximum transverse displacement and condition number of the stiffness matrix are given for each combination. Maximum displacement by Ritz reference (50×50 trial functions): $-4.43 \times 10^{-5} m$.

Activation	Range of weights	Range of bias	Diff. [%]	Cond. A
		$[-1,1]$	-37.92	2.18×10^{30}
	$[-1,1]$	$[-2,2]$	-37.02	2.09×10^{30}
		$[-3,3]$	-35.89	2.12×10^{30}
		$[-1,1]$	-43.34	8.73×10^{26}
Tanh	$[-2,2]$	$[-2,2]$	-41.31	1.75×10^{27}
		$[-3,3]$	-37.92	1.43×10^{28}
		$[-1,1]$	-38.15	3.70×10^{23}
	$[-3,3]$	$[-2,2]$	-41.98	6.43×10^{23}
		$[-3,3]$	-39.95	1.56×10^{24}
		$[-1,1]$	-53.72	5.73×10^{30}
	$[-1,1]$	$[-2,2]$	-53.05	5.12×10^{30}
		$[-3,3]$	-50.56	5.21×10^{30}
		$[-1,1]$	-39.27	1.82×10^{30}
Sigmoid	$[-2,2]$	$[-2,2]$	-37.92	1.57×10^{30}
		$[-3,3]$	-37.92	1.93×10^{30}
		$[-1,1]$	-41.76	7.11×10^{29}
	$[-3,3]$	$[-2,2]$	-41.98	5.28×10^{29}
		$[-3,3]$	-41.98	4.32×10^{29}

Table 22: Case of study on the influence of network hyperparameters for the case of a CFCF plate with *Configuration 1*. Variables: activation function, range of weights and range of bias. $\boldsymbol{A}$: stiffness matrix.

The reported results are not conclusive, but several considerations can be made from them. The maximum transverse displacement does not agree with the reference value for any tested configuration, it means the problematic persists independently of the network parameters. The condition numbers of the stiffness matrix are exceptionally high (MATLAB takes 10^{20} as a reference for badly conditioned matrices), what indicates the solver has a difficult time in finding the configuration that minimizes the norm of the output weights vector. This consideration cannot be taken hardly, and other sets of boundary conditions that do not include free edges present similar values of the condition numbers. No major differences are a priori observed between *tanh* and *sigmoid* functions, although the second one tends to underestimate furtherly the transverse displacement. Condition numbers are also higher for a sigmoid activation, therefore it has been opt to take *tanh* as default. It has been mentioned previously to be used in *Configurations 1,2,3*.

In view of the previous test cases, it is possible to conclude that there is not a single combination of parameters reported in this work that can alleviate the issues associated to the introduction of free boundaries. The applicability of different cases has been covered thoroughly, and it is possible to conclude the nature of this problem is numerical, and associated to the strong-form displacement based formulation used in this first collocation method, along with the network representation that is done of it. These considerations are also extensible to the context of free vibrations analysis, where finding the natural modes of a free-edge was hindered by this same problem. An alternative possibility has been studied in the form of a *mixed formulation*, although specific results are not reported. A Reissner-Mindlin theory alone was considered, shear forces (Q_x, Q_y) were introduced as additional variables of the simulation, along with two additional governing equations on shearing. The motivation for this change was to test a configuration were shearing was introduced explicitely into the problem and could be hardly imposed when required, forcing the system to get to a solution that enforces zero-shear at free edges. Both variables are treated along with displacements, network approximation includes specific output nodes and weights for them that have to be fitted during training. Although the

simulation became considerably heavier due to these extra parameters, an outstanding performance was achieved where transverse displacements at free edges is correctly reproduced. This formulation however is not straightforward to be extended into the compact notational form of SGUF, and a dedicated extension has been opened in the context of future works.

Test case 5 - influence of the number of neurons in free vibrations analysis

Regarding the free vibrations analysis, a considerably lower number of neurons (100) has been used as default in Section [5.2]. The motivation for this is introduced in Table [23], where the values of the first six natural frequencies are given in the context of plate *Configuration 1* for a decreasing number of neurons. The method uses the pseudoinverse preconditioner as default, results are compared with those from the Ritz method (50×50 trial functions) in order to compare frequency convergence. Boundary conditions: CCCC.

Case	**Mode [Hz]**					
	1	**2**	**3**	**4**	**5**	**6**
Present ($n = 500$)	476.91	1792.17	2307.17	3136.51	4388.19	7134.26
Present ($n = 400$)	1358.04	1823.53	2758.61	3748.62	4628.26	5838.39
Present ($n = 300$)	1726.34	3568.85	3736.55	4075.85	4645.77	5127.38
Present ($n = 200$)	2744.23	4251.19	5370.25	5589.57	9338.33	10920.92
Present ($n = 150$)	549.34	1497.85	1641.82	2043.81	2529.96	4278.69
Present ($n = 100$)	915.51	1854.86	1874.73	2856.91	5047.33	5219.92
Present ($n = 50$)	8143.39	18626.25	21424.04	32509.95	32782.90	35549.89
Ritz	880.34	1793.82	1793.82	2642.20	3211.71	3227.18

Table 23: First six natural frequencies convergence with decreasing number of neurons for a square plate with *Configuration 1*. Validation done with Ritz method solutions. Boundary conditions: CCCC.

Convergence is studied starting from 500 neurons, which is the default input in linear static analysis. Above it, natural frequencies quickly diverge, larger amounts of neurons are not considered. Results show a convergence in frequencies that is non-monotonic and some configurations shift them to high values. The setup with 100 neurons gives the most approximated results to reference for the first four modes, it was already reported in Table [24]. Beyond this configuration, for $n = 50$ neurons, the neural network has too few nodes and is unable to represent the situation accurately, first frequencies are missed and it converges to higher values. This analysis proves that the selection of hyperparameters has a large influence when it comes to determine natural frequencies, and the number of neurons is the most influential. Good spots are found for configurations with a small numbers of neurons, while a large number has the opposite effect.

Test case 6 - influence on the preconditioner in free vibrations analysis

In order to assess the performance of the free vibration tools, an extensive comparison on the performance of both preconditioning strategies mentioned in Section [4.4] has been done. This is a necessary first step to understand which of the two possibilities available resembles better the reference results. The first attempt to precondition the problem comes with the use of the stiffness matrix transpose, such that $\boldsymbol{P} = \boldsymbol{A}^{'}$. The setup of *Configuration 1* is used, with ED_{110} as the kinematic theory and SSSS boundary conditions of the plate. Network architecture is

reduced to the use of 100 neurons, which has shown the best performance. The transverse deflections of the first 16 modes are reported in Fig.(33), although the solver was only able to find the first thirteen.

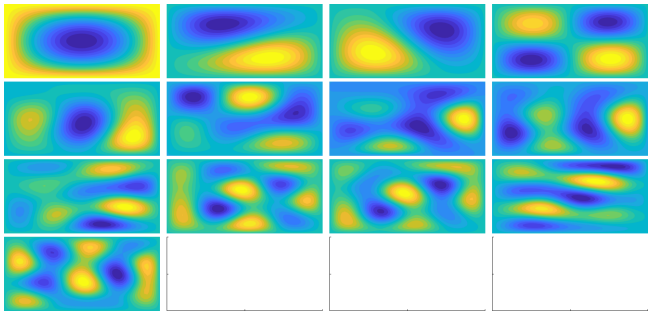
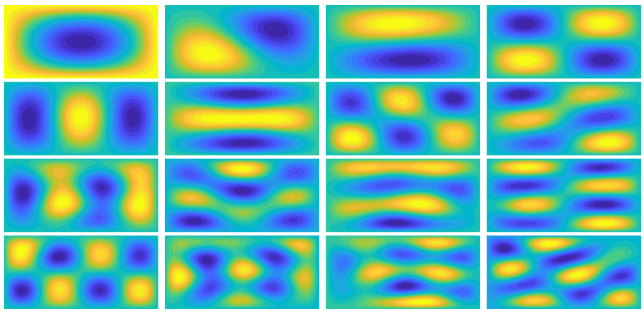


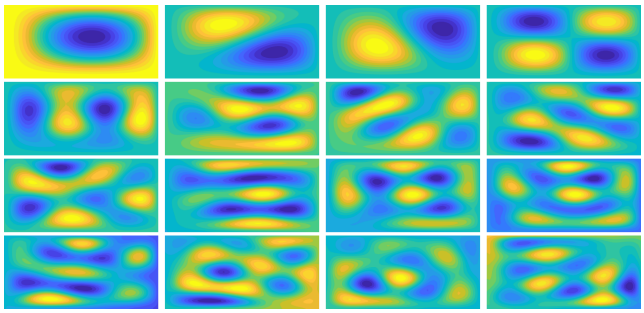
Figure 33: First 16 modes transverse displacement contours of a plate using a *transpose* preconditioner. Case setup from *Configuration 1*.

During evaluation, a higher number of eigenvalues is found, but only those real values are considered. The performance of this first strategy is bad, and spurious modes appear at intermediate frequencies with short half-wavelengths. From the numerical viewpoint, the stiffness matrix reports a very bad conditioning (on the order of 10^{23}, while $> 10^{20}$ already gives a warning from MATLAB to be an unrecommended simulation).

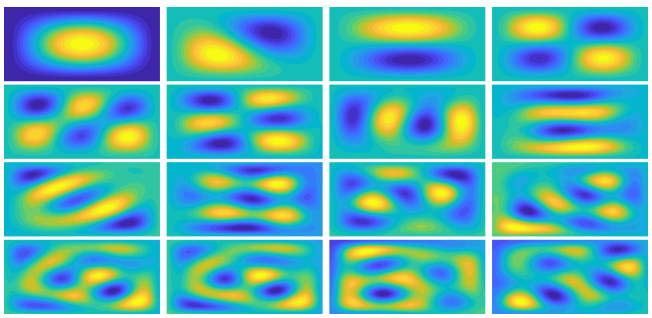
A second option has been studied, corresponding to a preconditioner of the *Moore-Penrose pseudoinverse* class. Fig.(34) shows the same comparison done in the previous case, for a set of different boundary conditions that implement this preconditioner.



(a) SSSS



(b) CCSS



(c) CCCC

Figure 34: First 16 modes of a plate in free vibrations using a *pseudoinverse* preconditioner for different boundary conditions. Case setup from *Configuration 1*. Vibration modes: a)SSSS, b)CCSS and c)CCCC.

It has proven a better performance compared to its predecessor in the approximation of the first few modes. The presence of spurious modes has been reduced and those with low values of the half-wave numbers are found with improved accuracy. The first six frequencies for the cases with both preconditioners are reported in Table [24] corresponding to Fig.(33) and (34). They have been validated against a FEM modal simulation which uses 30000 elements in a structured mesh.

Case	Boundary conditions	Mode	Present [Hz]	FEM [Hz]	Diff. [%]
$P = A'$	SSSS	1	483.20	481.94	-0.26
		2	1207.46	1205.90	-0.13
		3	1931.54	1205.90	-60.17
		4	2412.93	1926.20	-25.27
		5	2423.32	2413.00	-0.43
		6	2711.80	2413.00	-12.83
$P = A^{\dagger}$	CCCC	1	915.51	880.71	-3.95
		2	1854.86	1795.30	-3.32
		3	1874.73	1795.30	-4.42
		4	2856.91	2644.70	-8.02
		5	5047.33	3216.60	-56.92
		6	5219.92	3232.10	-61.50
	SSSS	1	483.17	481.94	-0.26
		2	1207.71	1205.90	-0.15
		3	1208.78	1205.90	-0.24
		4	1934.05	1926.20	-0.41
		5	2459.64	2413.00	-1.93
		6	2490.86	2413.00	-3.23
	CCSS	1	675.45	661.43	-2.12
		2	1506.76	1480.30	-1.79
		3	1514.64	1486.20	-1.91
		4	2354.19	2267.00	-3.85
		5	5272.55	2800.70	-88.26
		6	5645.44	2804.50	-101.30

Table 24: Validation of the first six natural frequencies against FEM modal simulation with the use of different preconditioners. Case setup from *Configuration 1*.

It is possible to appreciate the much better agreement in frequency in the cases where the pseudoinverse is used, what confirms its better applicability for this first collocation method compared to the transverse one. The quality of the prediction presents large differences among CCCC, CCSS and SSSS cases. 5% is considered an upper limit for validation, it is violated by the last modes of both CCCC and CCSS conditions, an indicative of the influence of clamped boundaries with respect to SSSS, where the difference is kept considerably lower for all modes. Therefore, those cases with clamped and simply-supported boundaries can be considered in the range of applicability of this analysis but their results must be carefully handled if more than two or three frequencies are considered. The pseudoinverse can be considered as well the best available tool to face this kind of analysis, transverse preconditioner is a priori discarded for future simulations.

Test case 7 - higher order kinematic theories in free vibrations analysis

The numerical assestment of higher order theories is explored in the context of the first collocation method. Dealing with high order kinematic theories has been impossible during validation, immediately higher theories than Mindlin predict a solution field for transverse displacement well away from validation results. Two cases with *Configuration 1* as their baseline

are considered in Fig.(35), using different sets of boundary conditions: CCCC and CCSS. Two high order kinematic theories are given, ED_{112}, ED_{332} compared to Mindlin theory in both cases. The contours of transverse displacement are shown in all three cases:

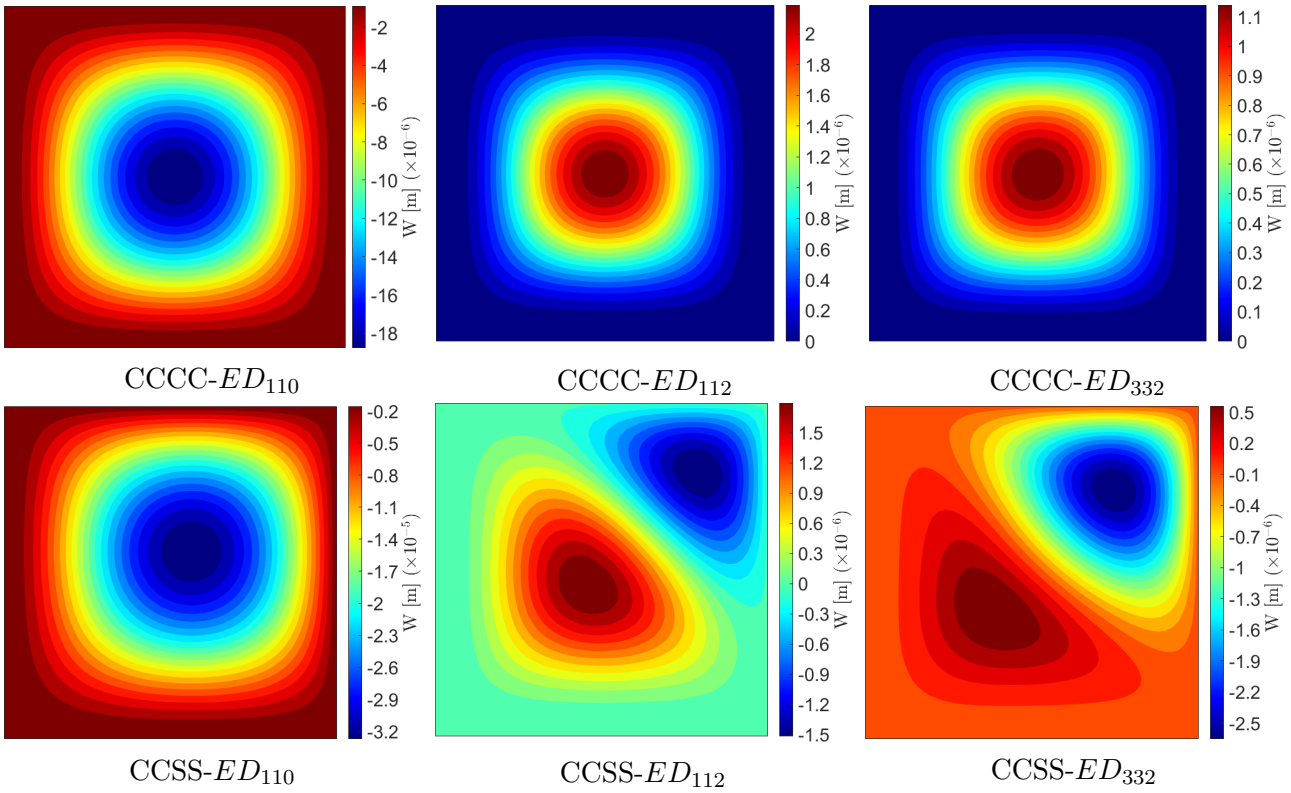


Figure 35: Contours of transverse displacement of plate Configuration 1 for different high order kinematic theories. Theory: ED_{110}, ED_{112}, ED_{332}. Boundary conditions CCCC (*top*) and CCSS (*bottom*). Linear static analysis.

For the CCCC case, the use of high order theories shows an interesting feature, the load applied is uniform and directed inwards from the contour (well represented in ED_{110} case, which has been previously validated), however, those theories are not capable of representing the deflection and predict that it goes outwards. With CCSS conditions, the deflection occurs in two directions and in two differentiated regions. It is important to compare these results with those of the same plate with SSSS boundaries in Table [5], where any higher order theory was shown to be able to represent the transverse displacement field. This behaviour indicates that the collocation method has a problem with constraining displacements in higher order theories, for which the values of the individual kinematic variables are set to zero at the boundary points. This new issue is added to the problems regarding plates with free edges, where dealing with high order theories is still not considered. The topic remains as an opened question for which further investigation is expected, and a solution must be provided.

5.5 Energy method - numerical assestment

Test case 8 - convergence study on the collocation grid size

The energy method has proven a more effective alternative than the first collocation method. Its natural ability to deal with high order kinematic theories and free edges has been proven accurate in most of the test cases covered during validation. However, certain specific issues remain opened and need of clarification from the numerical side. This approach uses the collocation grid in a very different way to the first method, a finer grid is needed. Those points

are not directly used to impose governing equations but in a numerical integration step that turns into a matricial problem thanks to the network approximation (topic covered in Section [4.3]). The number of points used during *training* influences the accuracy of the results, and may lead to an erroneous prediction of the displacement solution field. On the other hand, the extraction of results after training (displacements and second-order quantities) has proven to be influenced by a second grid size, chosen for interpolation between training points, in what is referred as *simulation*. Both parameters are subjected to a convergence study, starting from plate *Configuration 4*, where the strain energy is used as convergence parameter. A non-dimensionalization is used, the energy values are divided over its maximum in order to better appreciate the evolution in the graph, $\bar{U} = \frac{U}{max(U)}$. The second case has been obtained using the default $N = 100 \times 100$ training sampling grid from *Configuration 4*.

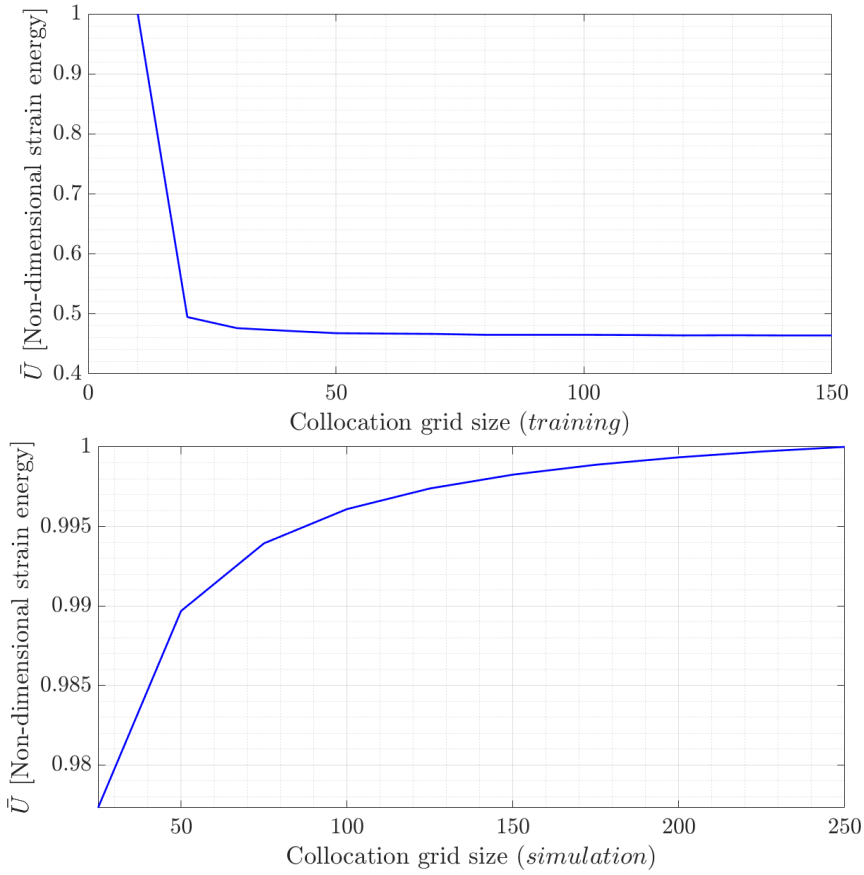


Figure 36: Convergence study over the strain energy. Different sizes of the collocation grid for training (*top*). Different sizes of the collocation grid for simulation (*bottom*). Boundary conditions: CCCC. Validity in both linear static and free vibrations analysis.

The value of the energy asymptotically converges to a constant value as expected, a grid $N = 100 \times 100$ is a reasonable sampling distribution (10000 samples) for training, in order to obtain an accurate integration. Smaller grids up to $N = 50 \times 50$ points (2500 samples) could also be used with a minimum loss in convergence. This statement holds for the considered square plate, where the error introduced by the area of approximation around integration points is kept to the minimum thanks to the uniform sampling technique. With more complex shapes, the sampling at the boundaries must be also considered (an improved boundary-sampling strategy was introduced in Section[5.2] to this respect), therefore using few points might prove insufficient. The first distribution involves a good trade-off between accuracy and computational complexity, reason why it has been used as a default in most weak formulation reference configurations, a sampling grid below $N = 50 \times 50$ is not recommended in any case. Furthermore, the

energy variation between simulation grids is minimal and any of them can be considered accurate enough (5% tolerance) for post-processing. The largest grid, with $N = 250 \times 250$ samples is considered a default for extracting results due to its minimum impact on computational time.

Test case 9 - convergence study on the number of neurons

The number of neurons used in the network hidden layer has been kept constant at 200 in every tested configuration. This criterion has been achieved regarding convergence of the train energy shown in Fig.(37) for an increasing number of neurons:

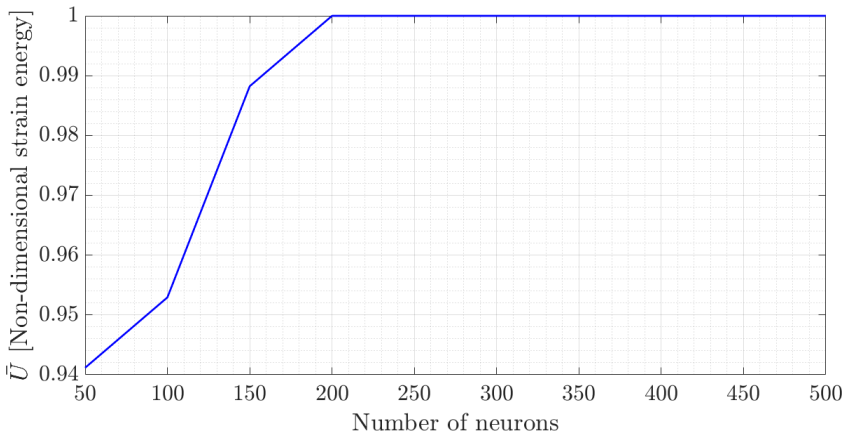


Figure 37: Convergence study on the strain energy. Changing number of neurons in the neural network hidden layer. Boundary conditions: CCCC. Validity in both linear static and free vibrations analysis.

The value of the energy arrives at an asymptotic value for 200 neurons, meaning the addition of more neurons does not improve the representation the network makes of the problem. This parameter has shown to be much less influential than it is in the context of the strong method. The convergence study done in Section [5.2] confirms this number of neurons can be taken as a good reference value, the solution field is accurate and according to validation. Taking more neurons into the network is quite inefficient from the computational side, it is the parameter most influencing the computational time, and if taken larger, higher order theories would become intractable with available computational resources. These considerations can be extended to orthotropic cases and any other setup, and can be considered a default value within this method. In opposition to the first collocation method, the energy method allows to use the same number of neurons both for linear static and free vibrations analysis, dropping the requirement to tune their value in order to improve the results.

Test case 10 - influence of different hyperparameters in the energy method

The influence that the selection of the activation function has on the solution is of paramout importance. This topic was previously covered for the first collocation method, the use of a different formulation requires the network hyperparameters be tested to better understand their internal behaviour. Plate from *Configuration 4* is considered in a test case, CFCF boundaries are used again to make a comparison possible with the results in Table [22]. *Tanh* and *sigmoid* are considered relevant activations for this analysis, ranges of weight and bias are varied in the same magnitudes. Results are reported in Table [25], where the difference with the reference

solution at the point of maximum transverse displacement is given, along with the condition number of the stiffness matrix. Ritz reference (50×50 trial functions): $-4.43 \times 10^{-5} m$.

Activation	Range of weights	Range of bias	Difference [%]	Cond. A
Tanh	$[-1, 1]$	$[-1, 1]$	-0.45	7.62×10^{22}
		$[-2, 2]$	-2.03	5.82×10^{21}
		$[-3, 3]$	-1.58	1.46×10^{21}
	$[-2, 2]$	$[-1, 1]$	-1.58	6.52×10^{21}
		$[-2, 2]$	-1.13	2.20×10^{22}
		$[-3, 3]$	-1.13	1.80×10^{21}
	$[-3, 3]$	$[-1, 1]$	-1.35	1.32×10^{19}
		$[-2, 2]$	-1.13	3.16×10^{20}
		$[-3, 3]$	-1.13	3.16×10^{20}
Sigmoid	$[-1, 1]$	$[-1, 1]$	-4.06	6.68×10^{22}
		$[-2, 2]$	-3.84	3.62×10^{22}
		$[-3, 3]$	-4.74	6.17×10^{23}
	$[-2, 2]$	$[-1, 1]$	-2.03	8.86×10^{20}
		$[-2, 2]$	-2.03	2.35×10^{21}
		$[-3, 3]$	-1.58	3.64×10^{21}
	$[-3, 3]$	$[-1, 1]$	-2.03	1.13×10^{21}
		$[-2, 2]$	-1.80	5.04×10^{21}
		$[-3, 3]$	-1.80	8.53×10^{20}

Table 25: Case of study on the influence of network hyperparameters for the case of a CFCF plate with *Configuration 4*. Variables: activation function, range of weights and range of bias. $\boldsymbol{A}$: stiffness matrix.

Reported differences are within admissible tolerance of 5% in all cases. The agreement with the reference is much better than with the first collocation method, the energy approach performs better in approximating the transverse displacement field, independently of the hyperparameters configuration. A slightly bigger error is observed for *sigmoid* activation with the empirical $[-1, 1]$ range of weights, the stiffness matrix condition number is also bigger in this case, indicating it is specially badly conditioned. In a general sense, the condition number is several orders of magnitude lower than those cases reported in Table [22], but still ill-conditioned, and changing parameters does not have a great effect to this respect. Modifying the range of bias has shown the same tendency of its strong method counterpart, it is not an influential parameter that can be left at its empirical range $[-1, 1]$.

Major differences in the solution due to modifications in network parameters have been reported in free vibrations analysis in Section [5.3]. Therefore, comparing both activations to approximate the natural frequencies of the panel makes a better statement on which one shows an improved behaviour for this problem. Fig.(38) introduces the evolution of the first six natural frequencies in the previous case, with different range of weights and both activation functions:

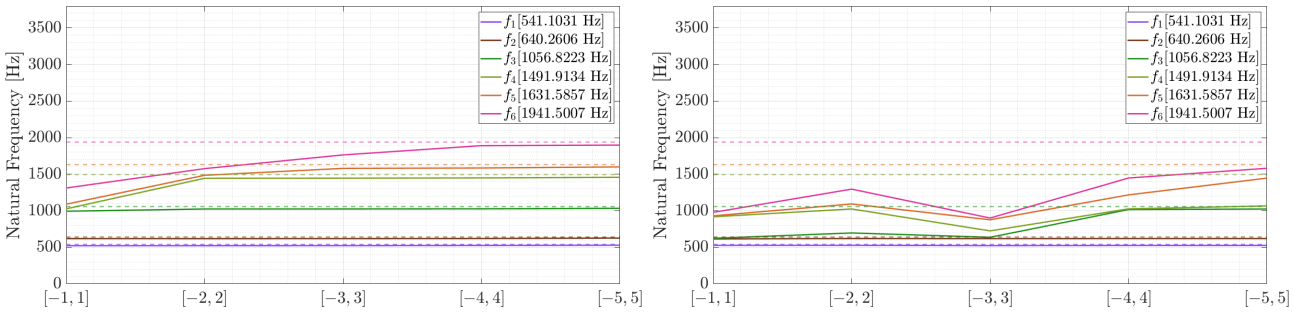


Figure 38: Convergence of first six natural frequencies for different range of weights and activation functions. Hyperbolic tangent activation (*left*), sigmoid activation (*right*). Boundary conditions: CFCF.

It is important to remark in the first place that range of weights at which converged frequencies better resemble reference values is entirely case dependent. For instance, Table [26] shown a validation of natural frequencies for different boundary conditions, where the best spot in terms of modified range of weights was found at $w = [-3, 3]$. Nevertheless, tendencies in convergence are the same and are mostly influenced by the activation function. In the case of hyperbolic tangent activation, frequencies converge from below and reach a point within problem tolerance (5%) for $w = [-5, 5]$. This convergence is monotonous, those mode shapes of transverse displacement with short half-wavelengths appearing at intermediate frequencies, are progressively screened out for increased ranges of weights. On the other hand, sigmoid activation is not capable of reaching convergence except in the first two frequencies. Therefore, it can be concluded that *tanh* activation works better in approximating the eigenproblem than *sigmoid*, but these differences are stretched in the context of a linear static analysis. Regarding these considerations, *tanh* activation has been considered a default selection in all energy method configurations. Further studies would be required to better understand why these differences between both kinds of analyses are present.

Test case 11 - influence of the penalty factor

The influence of the penalty factor on the solution can be considered the most delicate parameter within the energy method. Its selection is totally case dependent, boundary conditions, geometry or single-layered/multilayered configurations have an effect on finding the factor that ensures results to be within admissible tolerance with respect to references. Each simulation considered in Section [5.3] required of a specific penalty factor along with the rest of parameters, which had to be obtained by trial and error, making the problem to be computationally costly and time consuming, specially when high order models and multiple layups are computed. Regarding the linear static analysis, the behaviour of the solution field with a too low or too large penalty is to overestimate or underestimate the transverse displacements, respectively. In a free vibrations analysis, its behaviour can be followed more clearly. An example of it is given in Fig.(39), where the panel from *Configuration 4* is analysed with a wide range of penalty factors. The evolution of the first six natural frequencies is given for two sets of boundary conditions, CCCC and FCFC. Even though it has been said the behaviour introduced by the penalty is case dependent, it is relevant to compare two cases in which one of them has several free boundaries, the penalty contribution will include less kinematic variables in its definition. Results from a Ritz simulation (50×50 trial functions) are used to compare the values computed and identify the regions where the penalty factor performs better.

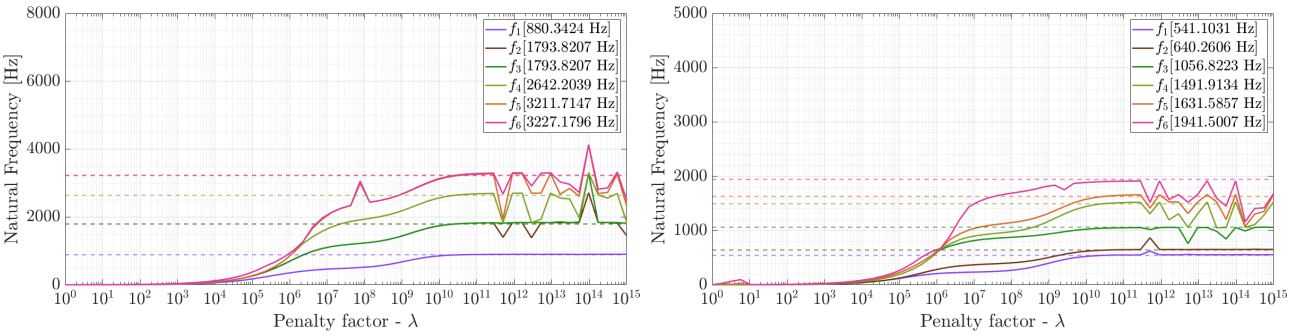


Figure 39: Convergence of the first six natural frequencies with respect to penalty factor. Boundary conditions: CCCC (*left*) and FCFC (*right*).

Three regions of interest are differentiated: between $10^0 - 10^{10}$ the penalty factor is low and frequencies are generally underestimated. A plateau region where the prediction reaches reference values is observed in the range $10^{10} - 10^{11}$. It is in this region where all cases run with *Configuration 4* have had their penalty chosen (with minor variations). A third region spans $10^{11} - 10^{15}$ and above, the penalty factor is too large and a spurious effect is observed in the form of an erratic evolution of natural frequencies. A similar analysis can be derived from both cases, the expected differences in a less constrained plate are not reproduced, the influence of the boundary conditions is not an obstacle, what makes the method more robust for the study of different configurations. Finding an optimal way to determine the value of the penalty factor independently of the case considered remains an opened topic that must be adressed in future works. From the experience of this study, the optimum penalty region has been found to coincide typically with the order of magnitude of the diagonal elements in the stiffnes matrix prior to the assembly of the energy contribution. It can be considered a useful rule-of-thumb to initialize this parameter.

An additional test case has been considered but is not reported in this thesis. It consists on the use of the *method of Lagrange multipliers* to introduce Dirichlet conditions into the minimization of the total potential energy. Those Lagrange multipliers take part of the network approximation as additional variables to solve for, releasing the need to manually select the value of the penalty. This approach has shown however two major problems that justify why it has not been included in the analysis: although the multipliers introduce constraints at equation level, the initialization of these parameters is of a very different magnitude than the terms in the strain energy matrix. It becomes necessary to introduce the characteristic length (see Section [4.3]) to premultiply Lagrange multipliers, but this step is equivalent to using penalty factors, the characteristic lengths must be tuned manually. Additionally, the method generally underestimates the transverse displacement fields for any boundary conditions. Further study on this topic is required to make this strategy suitable to be used within the energy method.

Test case 12 - influence on the preconditioner in free vibrations analysis

The free vibrations analysis in the context of the weak-form formulation method faces the same problematic of its predecessor. Raw evaluation of the eigenproblem without a proper preconditioning step leads to a complete misprediction of the natural modes. The same strategy is tested, the use of the stiffness matrix pseudoinverse as preconditioner, that gave excellent results in previous cases. Computation of the eigenvalues either with no preconditioner or a transpose-kind one lead to erroneous predictions of the modes and have been omitted from the analysis.

Plate setup in *Configuration 4* is used to validate the first six frequencies of several cases with different sets of boundary conditions against a FEM simulation used as reference, in Table [26]. A third comparison is introduced with a non-preconditioned configuration that uses a modified network range of weights of $w_i = [-3, 3]$, in a similar manner of the strategy used in the convergence study. The motivation of this test case is to quantify the actual difference between using both possibilities and their deviation to reference values. Due to the good convergence in frequency and the simplicity of the configuration (isotropic and single layered alluminum), the theory chosen is ED_{110}.

Boundary conditions	**Mode**	**Present-$A^\dagger$ [Hz]**	**Present-Modified w [Hz]**	**FEM [Hz]**
CCCC ($\lambda = 10^{10}$)	1	848.67	888.71	880.71
	2	1735.24	1808.46	1795.30
	3	1739.38	1812.27	1795.30
	4	2506.81	2661.14	2644.70
	5	3114.69	3241.13	3216.60
	6	3220.16	3257.32	3232.10
SSSS ($\lambda = 10^{10}$)	1	486.01	487.29	481.94
	2	1211.46	1214.01	1205.90
	3	1211.94	1214.07	1205.90
	4	1940.70	1946.88	1926.20
	5	4008.98	2420.21	2413.00
	6	4111.82	2420.33	2413.00
CCSS ($\lambda = 10^{10}$)	1	652.36	667.64	661.43
	2	1459.62	1490.41	1480.30
	3	1467.41	1498.10	1486.20
	4	2243.43	2286.74	2267.00
	5	2755.22	2815.72	2800.70
	6	2788.10	2819.56	2804.50
CCCF ($\lambda = 10^{10}$)	1	549.49	587.79	582.98
	2	957.33	973.12	974.08
	3	1903.81	1554.78	1542.10
	4	2017.59	1860.81	1871.90
	5	2889.73	1970.64	1962.80
	6	3550.05	2804.83	2840.30
SSSF ($\lambda = 10^{10}$)	1	284.46	284.51	283.30
	2	675.48	674.33	673.51
	3	1004.59	1005.03	1003.90
	4	1442.36	1440.88	1435.30
	5	1497.70	1498.42	1507.70
	6	2184.08	2202.83	2202.90

Table 26: Validation of the first six natural frequencies against FEM modal simulation for two eigenproblem strategies. 1) Pseudoinverse preconditioner of the stiffness matrix, 2) Modified range of weights. Case setup from *Configuration 4*.

Most frequencies lay in the validation range with reference results. Only differences are observed in higher modes and regarding the use of the pseudoinverse preconditioner. The modified weights strategy outperforms its predecessor, specially at higher frequencies where it is capable of discarding spurious modes with shorther half-wavelengths appearing at intermediate frequencies. These results demonstrate that this strategy can be considered a preferred option to solve the eigenproblem within the energy method, although it introduces the complexity of weight range limit as an additional parameter.

5.6 Sensitivity analysis

A deeper analysis is performed regarding the coupled interaction of different hyperparameters. This study extends only to the case of the energy method, which has shown a better applicability in structural problems. Previous chapter covered the numerical assestment of different issues associated to each method, as well as the influence of the selection of network parameters on the solution. In this chapter, that dependence is extended and those parameters are modified in pairs in order to identify if and how they are correlated. The second motivation is to find spots in these combinations that lead to a generalized improvement of the results.

All test cases consider plate *Configuration 4*, except for those parameters specifically varied in the analysis. Validation is performed against a Ritz simulation (50×50 trial functions).

Hyperparameters - Collocation grid size vs. number of neurons

Size of the collocation grid and number of neurons are the two parameters with the most influence over the computational size of the problem. Considering them alone, a default point of operation has been selected in Section [5.5] for 100×100 sampling points and 200 neurons, which has shown a good performance in the majority of cases tested. Both parameters are modified together in order to explore a larger region of validity, where the deviation from the reference solution is minor. A contour plot of the percentual difference of maximum transverse displacement is shown in Fig.(40).

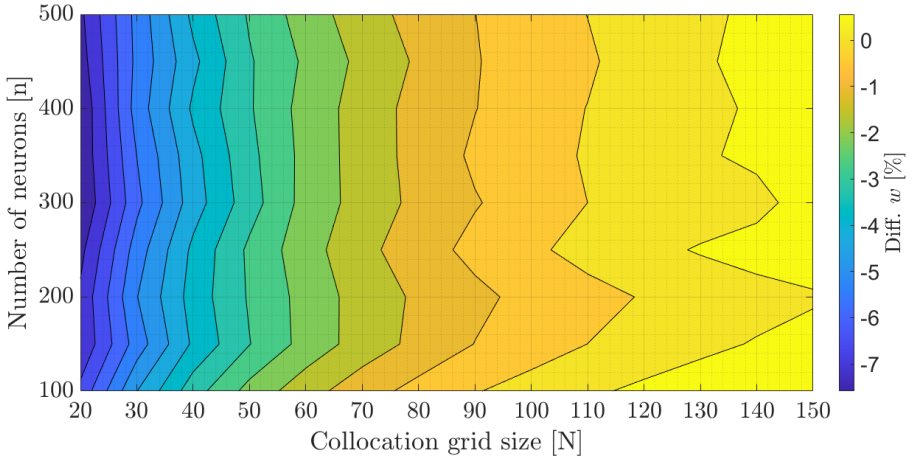


Figure 40: Collocation grid size vs. number of neurons case study, percentual difference at maximum transverse displacement with reference solution. Reference: $1.89 \times 10^{-5} m$

In general, selection of the collocation grid dominates over the number of neurons, although both parameters are not totally independent. A small grid tends to underestimate the displacement field, this difference becomes smaller towards larger grids with minor influence of the second parameter. On the other hand, the magnitude of the difference is within admissible tolerance (5%) for the majority of results, meaning a small collocation grid on the order of 50 points per direction is still an acceptable choice. The latter is an important consideration, provided certain cases of study in Section [5.3] employed a smaller grid to make them computationally feasible. This test case justifies that choice, but it must be also considered that the difference might be larger for a different panel geometry, and should be tested in correspondence. The default point lies in a strip with a minimal difference, bounded by 1% deviation, other configurations are also possible with more and less neurons and collocation points. It is relevant to observe that with 100 neurons, 80-90 points per direction are enough to obtain a similar performance

with a considerably lower computational cost, although this number of neurons is considered as insufficient in other cases.

Hyperparameters - Collocation grid size vs. penalty factor

The selection of an optimum value for the penalty factor is a recurrent topic in the context of the energy method. The number of points used by the collocation grid is considered along with the penalty factor. A contour plot of the percentual difference of maximum transverse displacement is shown in Fig.(41):

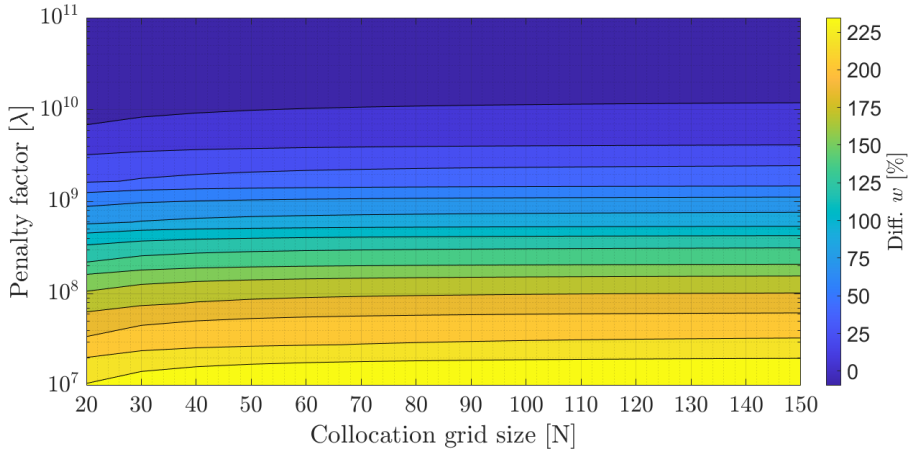


Figure 41: Collocation grid size vs. penalty factor case study, percentual difference at maximum transverse displacement with reference solution. Reference: $1.89 \times 10^{-5} m$

These parameters are uncorrelated. The number of sampling points determines how accurately numerical integration is done, but it has no influence on the order of magnitude of the components of the stiffnes matrix. Those components are partially penalized by the penalty factor in the corresponding kinematic variables constraint at the boundary regions. Therefore, the influences of both items are completely independent, which is proven by the graph. A special consideration can be made by looking at the first values of the grid size, the level lines converge from below at very coarse grids. This can be understood as an insufficient sampling of the grid which cannot properly map the geometry, and not as a combined effect of both parameters.

Hyperparameters - Collocation grid size vs. range of weights

This test case considers the collocation grid size and the range of weights are modified together in order to explore their correlation. A contour plot of the percentual difference of maximum transverse displacement is shown in Fig.(42):

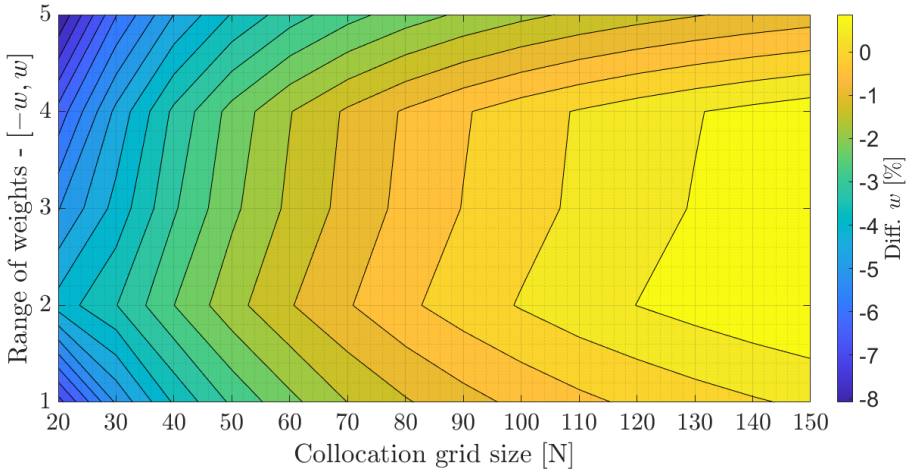


Figure 42: Collocation grid size vs. range of weights case study, percentual difference at maximum transverse displacement with reference solution. Reference: $1.89 \times 10^{-5} m$

Both parameters are shown to be related in an unclear way. A coarse collocation grid is associated to a larger difference with respect to reference displacement, the finer it becomes, the difference reduces. On the other hand, the range of weights improves the solution in an intermediate range ($[\pm 2, \pm 4]$) for any collocation grid size. It is an indicative that modifying the weight range can be benefitial in the context of linear static analysis, although it has not been extensively used in previous chapters in this respect due to its inherent case dependency. The largest values of the range of weights generally worsen the solution due to the effect of many neurons approaching saturation values. The representation capabilities of the network become hindered, making the solution to move away from the reference value.

Hyperparameters - Number of neurons vs. penalty factor

The number of neurons is studied in comparison with the penalty factor. In Fig.(43), a contour plot of the percentual difference of maximum transverse deflection is shown in order to understand the correlation between both parameters:

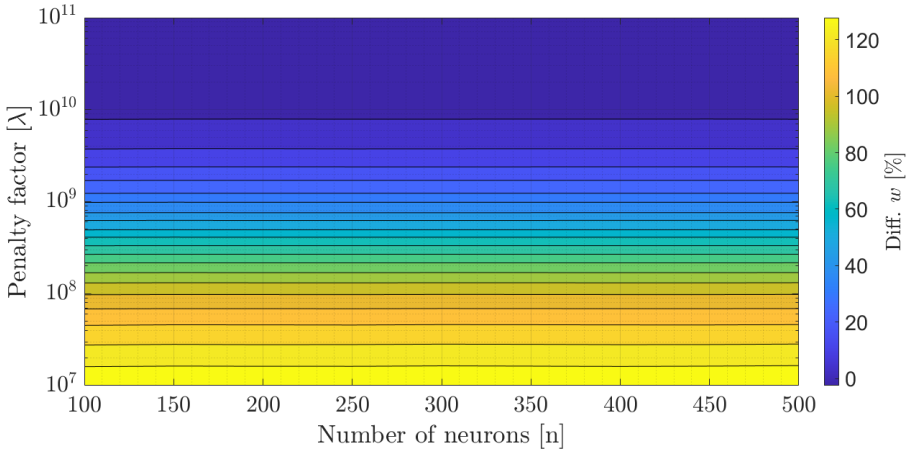


Figure 43: Number of neurons vs. penalty factor, percentual difference at maximum transverse displacement with reference solution. Reference: $1.89 \times 10^{-5} m$

The straight level lines are an indicative that these two parameters are not introducing any coupling effects into the solution. It can be explained from the perspective of the neural network approximation, where the number of neurons is introducing rows and columns that represent the values of activated nodes. Those kinematic variables represented by the neurons

are used along with the governing equations to set up the assembled stiffness matrix, therefore the components of that matrix are of the same order of the quantities used in the equations (constitutive constants, kernels...). On the other hand, the penalty factor is artificially inflating the values of those constraint degrees of freedom corresponding to essential boundary conditions, that are assembled as well. Using a fewer or larger number of neurons influences only the size of the stiffness matrix but not the order of its components, therefore the penalty factor does not need to be modified accordingly in any case. These considerations allow to release the dependence of the parameters, that can be taken independently.

Hyperparameters - Range of weights vs. penalty factor

The influence of the range of weights has been explained to be of paramount importance regarding the free vibrations analysis with the energy method. It has to be manually selected by trial and error, as well as the penalty factor. Therefore, it is important to understand how they both are correlated and if one selection influences the other. Fig.(44) shows a contour plot of the percentual difference of the maximum transverse displacement with respect to reference value.

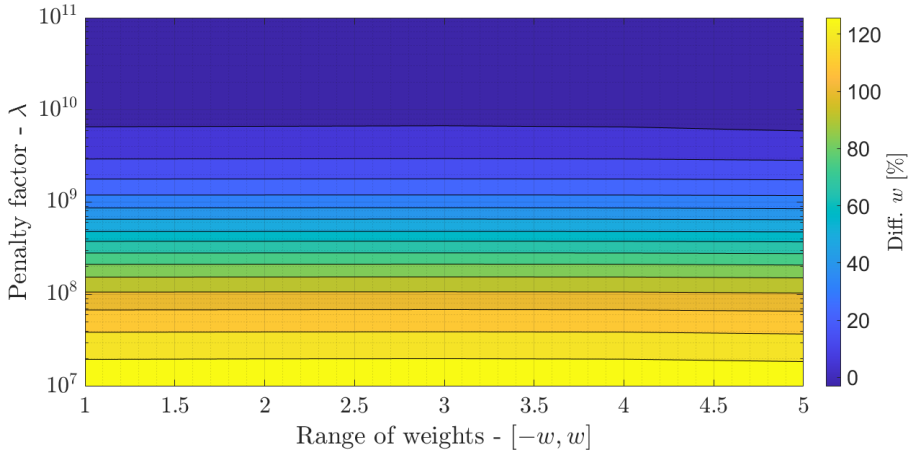


Figure 44: Range of weights vs. penalty factor, percentual difference at maximum transverse displacement with reference. Reference: $1.89 \times 10^{-5} m$

It can be stated that both parameters are a priori uncorrelated. Following the same reasoning of the study between number of neurons and penalty factor, both parameters affect the linear problem in a different way. Regarding the use of activation functions, hyperbolic tangent has been specified as default in the energy method due to its better performance (see Section [5.5]). The range of weights determines the values that the connections between input and hidden layer can take in the initial generation, those values multiply the input data and then the biases are added to determine the input value of the activation function. When the range of weights is enlarged, those inputs become statistically bigger meaning they are more prone to reach saturation values (extremes) of the activation function, in this case they tend to $1/-1$ as long as the input approaches infinity. This fact influences the values of the activated nodes in the network approximation, but not their order of magnitude which is given by the physics of the problem. Therefore, the penalty factor does not need to be tuned accordingly.

An additional effect can be appreciated by looking at the tendency of the upper level line in the contour, which goes down for the largest range of weights. This is a parasitic effect that takes place whenever a too high range is considered. Many neurons are activated near saturation, hindering the representation that the network makes of the physical case. The performance is

poor in those cases, that must be avoided.

5.7 Computational assestment

All computations have been performed in an *Acer F5-573G* laptop with *Intel Core(TM) i7-7500U @2.70GHz,16 Gb RAM* and *NVIDIA GeForce GTX 950M* graphic card. Major technical limitations are given by RAM memory, provided a large capacity is required to temporarily store assembled matrices, and by CPU clock-rate, that directly influences code execution speed. Both cases in *Configuration 1* and *Configuration 4* are tested in a convergence study that involves three parameters: collocation grid size, number of neurons, and kinematic theory. All of them play a primary role in the need for computational resources. Fig.(45) shows the code run time and peak-memory usage of these two configurations in the available hardware, which are used to exemplify technical difficulties:

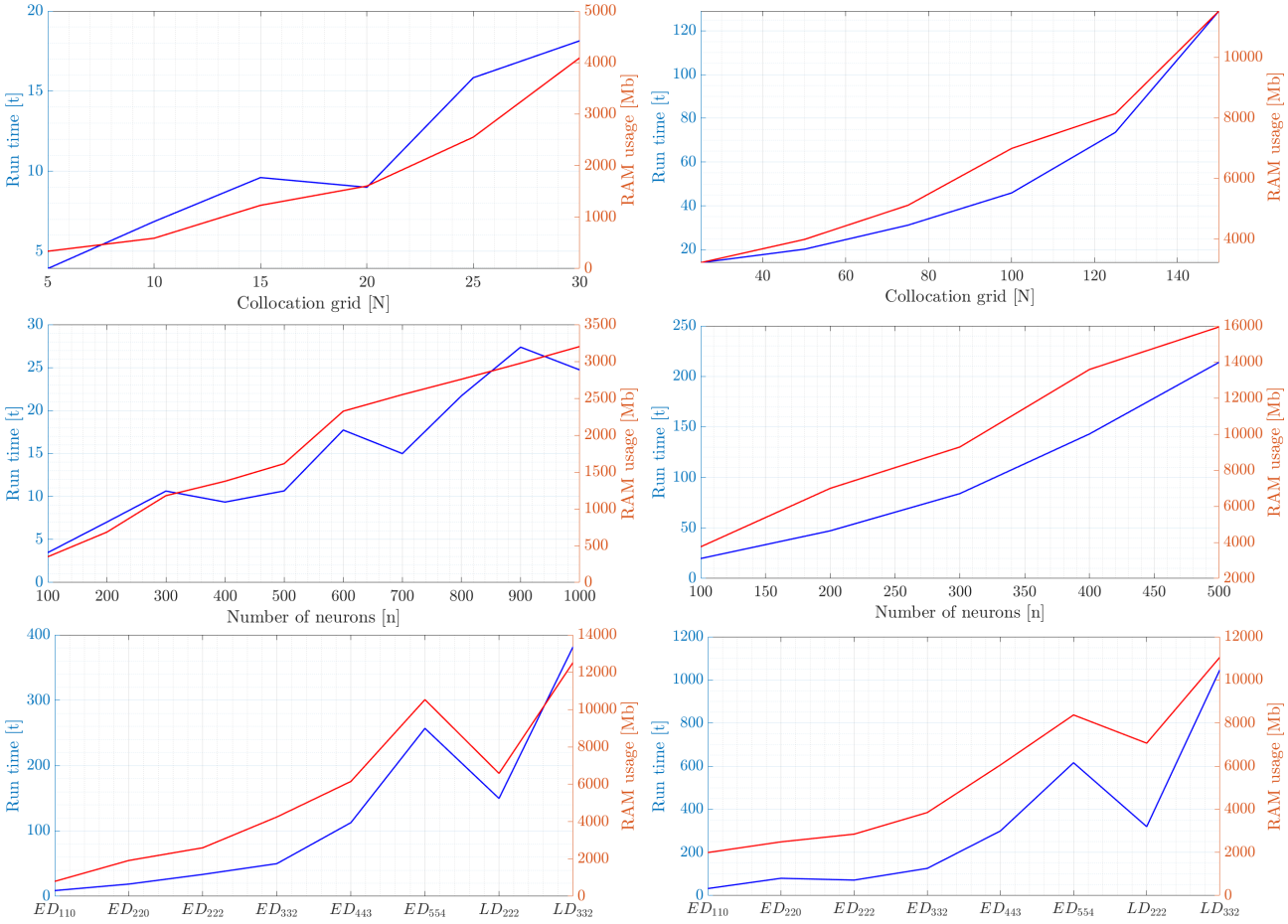


Figure 45: Computational performance assestment of collocation methods against different parameters and models. Strong-form collocation method (*left*), and energy method (*right*). Computational time includes training and simulation, while RAM usage is measured at peak execution.

Run time has proven one order of magnitude lower in the case of the strong collocation method, the faster assembly of the stiffness matrix iteratively adds the governing equations in the rows direction of the matrix. In the energy method, full matrices are handled and added together, each one representing the energy contribution of a given kinematic variable. These long matricial operations take a longer execution. RAM usage is difficult to compare, given the different values used in the hyperparameters. Generally, the energy method requires a larger amount

of RAM capacity at its peak operation point, due to the temporary storage of matrices prior to assembly. Nevertheless, memory usage drops to minimum level during training, as it only requires to store the matrices for the linear problem, which grows depending only on the number of neurons. For the strong method, memory usage is in general lower, but even a minor increase in the size of the collocation grid involves a large growth in the size of the assembled matrices. Possible improvements to these issues are specified in Section [7].

6 Conclusion

The presented work has regarded the development of a robust computational methodology for the linear analysis of plates in the context of the *Sublaminate Generalized Unified Formulation* within a PINN-ELM framework. The development has dealt with different approximations to the problem, namely referred as *strong-form* and *weak-form* formulations, and their implementation in two collocation methods. Moreover, the solution capabilities have been eventually extended in the context of shells and general geometry plates, including holed panels and those with curved boundaries.

Starting from the PVW statement in terms of internal and external virtual work equivalence, a set of governing equations has been derived in the context of the strong collocation approach, along with a family of Neumann boundary conditions. This first methodology has proven satisfactory in low order kinematic theories used to describe the displacement fields within the linear static analysis, but subjected to certain unidentified numerical problems that extremely limit its capabilities. These issues appear when free edge conditions are imposed, and no improvement has been reported with the use of higher order theories. Additionally, its wrong of the transverse displacement field corrects for very thick configurations, when shear effects are undoubtedly present. The analysis has been extended to the field of free vibrations, where the first collocation method has shown a poor performance to approximate the natural frequencies, even with the use of specific preconditioners.

In order to overcome these limitations, an energy method based on the weak-form formulation has been proposed, aimed at minimizing the value of the total potential energy. The elastic strain energy and the external energy contributions are recovered from the PVW statement by imposing the stationarity conditions on the total potential energy, and evaluated numerically thanks to a numerical integration scheme. This approach has shown a better performance over the strong collocation method, where the description of free edges is naturally solved due to the intrinsic enforcement of natural conditions within the formulation. Dirichlet-type boundaries, on the other hand, are introduced with a penalty contribution. The latter becomes the major bottleneck of this approach, and a careful tuning of this parameter is required to achieve an accurate solution. A numerical assestment study has been performed in the context of both methods in order to justify the selection of certain ranges of parameters. Additionally, the crossed-influence of the combination of different network hyperparameters along with penalty factors and collocation grid sizes has been measured with a set of sensitivity studies. This analysis gives a clearer picture on the interdependence of those factors and how they must be dealt with depending on the case at hand.

The main achievements of this work can be summarized as: (i) succesful implementation of a PINN-ELM framework for the analysis of the linear behaviour of plates in a MATLAB code, (ii) extended capabilities to work with shells and general geometry panels, (iii) generation of two different approaches to the problem with two independent formulations and (iv) extensive testing of the parameters involved within the neural network computations and their influence on the problem.

7 Further improvement

This project is meant to open a new line of study on the application of Machine Learning to solve structural problems. PINN architectures have been resported used by previous authors within *deep learning* strategies, while ELM makes the current element of novelty. However, this new implementation has found several issues throughout the way that remain at least partially opened and should be covered in future works.

- **Free-edge numerical problematic:** principal issue associated with the strong formulation, it limits widely the capabilities of this method. A future approach to this problem must be twofold, extension of the mixed formulation to the context of SGUF, and further testing of the method without ELM, inside a traditional PINN network architecture.
- **Poor prediction of eigenvalues:** in the context of the strong formulation, results of the free vibrations analysis have been shown to be poorly predicted by the collocation method. Improving conditions have been added in the form of *preconditioning*, which has a positive effect for low order theories, although they are incapable of performing accordingly for high order theories.
- **Selection of an optimal penalty factor:** the biggest problematic of the energy method is to properly select the value of the penalty to use in the computation. A penalty either too high or too low has been proven to either overstiffen the problem or to loosely enforce essential conditions, leading to wrong predictions of the displacement field. This parameter is extremely sensitive, and completely problem dependent, what makes it a difficult task to develop a proper rule. Either a method for its selection or an alternative to the enforcement of Dirichlet-type conditions must be seeked as the next step for the development of this approach.
- **Demanding computational power:** regarding its computer implementation, both approaches covered present two essential problems, a large runtime (in the order of minutes) that becomes a concern for higher order models, and a high-demand of RAM capacity to store intermediate results (usually exceeding 16Gb and 32Gb storage capacities). Both problematics are expected to improve by coding these intermediate operations on a low-level programming language such as C++, which can be interfaced with MATLAB via MEX functions. Regarding the runtime, it is also recommended to derive large matrix operations to be executed by dedicated GPU arrays, which have been tested as an alternative but are not code-wise implemented.

References

[1] C.S. Kale, S. Gopalacharyulu, and B.S.R. Rao. Analysis of a clamped skew plate under uniform loading. *AIAA Journal 10*, pages 1–10, 1972.

[2] N. Fantuzzi and F. Tornabene. Strong Formulation Isogeometric Analysis (SFIGA) for laminated composite arbitrarily shaped plates. *Composites Part B 96*, pages 1–5, 2016.

[3] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, and J.M. Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research 18*, pages 1–13, 2018.

[4] D.K. Rajak and D.D. Pagar. Recent progress of reinforcement materials: a comprehensive overview of composite materials. *Journal of Materials Research and Technology*, pages 6–10, 2019.

[5] J. Pora. Composite materials in the Airbus A380: from history to future. *Airbus Large Aircraft Division*, pages 1–5.

[6] A. Crosky and G. Pearce. *Advances in Composites Manufacturing and Process Design.* Woodhead Publishing, 1st edition, 2015.

[7] P.M. Wegner, J.E. Higgins, and B.P. VanWest. Application of Advanced Grid-Stiffened structures technology to the Minotaur payload fairing. *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pages 41–44, 2002.

[8] Z.S. Toor. Space applications of composite materials. *Journal of Space Technology*, pages 1–3, 2018.

[9] Klaus Friedrich and Abdulhakim A. Almajid. Manufacturing aspects of advanced polymer composites for automotive applications. *Applied Composite Materials 20*, pages 1–11, 2012.

[10] J.N. Reddy and D.H. Robbins. Theories and computational models for composite laminates. *Applied Mechanics Reviews 47*, pages 147–169, 1994.

[11] A.K. Noor and W.S. Burton. Assestment of computational models for multilayered composite shells. *Applied Mechanics Reviews 43*, pages 67–97, 1990.

[12] J.N. Reddy. An evaluation of equivalent single layer and layerwise theories of composite laminates. *Composite Structures 25*, pages 21–35, 1993.

[13] E. Carrera. Historical review of zig-zag theories for multilayered plates and shells. *Applied Mechanics Reviews 56*, pages 272–305, 2003.

[14] S. Brischetto. *Ph.D Dissertation: Classical and mixed multilayered plate/shell models for multifield problems analysis.* PhD thesis, Politecnico di Torino, 2009.

[15] E. Oñate. *Thick/Thin Plates. Reissner-Mindlin Theory.* Springer, 1st edition, 2013.

[16] M. D'Ottavio. A Sublaminate Generalized Unified Formulation for the analysis of composite structures. *Composite Structures 142*, pages 187–199, 2016.

[17] A. Szekrényes. Application of Reddy's third-order theory to delaminated orthotropic composite plates. *European Journal of Mechanics A/Solids*, pages 2–3, 2013.

[18] H.G. Allen. *Analysis and design of structural sandwich panels*. Pergamon Press, 1st edition, 1969.

[19] J.N. Reddy. A generalization of two-dimensional theories of laminated composite laminates. *Communications in Applied Numerical Methods 3*, pages 173–180, 1987.

[20] E. Carrera, M. Cinefra, E. Zappino, and M. Petrolo. *Finite Element Analysis of Structures through Unified Formulation*. Wiley, 1st edition, 2014.

[21] L. Demasi. Mixed plate theories based on the Generalized Unified Formulation. *Journal of Composite Structures 87*, pages 1–3, 2008.

[22] L. Demasi. Partially Layer Wise advanced zig-zag and HSDT models based on the Generalized Unified Formulation. *Engineering Structures 53*, pages 1–6, 2013.

[23] E.Carrera. Theories and Finite Elements for multilayered, anisotropic, composite plates and shells. *Archives of Computational Methods in Engineering 9*, pages 29–37, 2002.

[24] J. Kiendl, F. Auricchio, L. Beirao da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for Reissner-Mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering 284*, pages 3–10, 2014.

[25] G.S. Pavan and K.S. Nanjunda Rao. Bending analysis of laminated composite plates using isogeometric collocation method. *Composite Structures 176*, pages 6–14, 2017.

[26] F. Tornabene, N. Fantuzzi, F. Ubertini, and E. Viola. Strong formulation finite element method based on differential quadrature: a survey. *Applied Mechanics Reviews 67*, pages 1–5, 2015.

[27] F. Tornabene, N. Fantuzzi, and M. Bacciocchi. Finite elements based on strong and weak formulations for structural mechanics: stability, accuracy and reliability. *International Journal of Engineering and Applied Sciences*, pages 1–21, 2017.

[28] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Elsevier Ltd., 6th edition, 2005.

[29] O.O. Ochoa and J.N. Reddy. *Solid Mechanics and Its Applications Volume 7: Finite Element Analysis of Composite Laminates*. Springer Science, 1st edition, 1992.

[30] J. Han and S.V. Hoa. A three-dimensional multilayer composite finite element for stress analysis of composite laminates. *International Journal for Numerical Methods in Engineering 36*, pages 1–11, 1993.

[31] T.J.R Hugues, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering 194*, pages 1–5, 2005.

[32] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R Hugues. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering 195*, pages 1–24, 2005.

[33] T.J.R. Hugues, J.A. Evans, and A. Reali. Finite element and NURBS approximations of eigenvalue, boundary-value and initial-value problems. *Computer Methods in Applied Mechanics and Engineering 272*, pages 1–5, 2013.

[34] J.A. Cottrell, T.J.R. Hugues, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering 196*, pages 1–13, 2007.

[35] V. Gulizzi, I. Bendetti, and A. Milazzo. A high-resolution layer-wise discontinuous galerkin formulation for multilayered composite plates. *Composite Structures 242*, pages 1–6, 2020.

[36] X.L. Chen, G.R. Liu, and S.P. Lim. An element free Galerkin method for the free vibration analysis of composite laminates of complicated shape. *Composite Structures 59*, pages 1–4, 2003.

[37] A.J.M. Ferreira, L.M.S. Castro, and S. Bertoluzza. A high order collocation method for the static and vibration analysis of composite plates using a first-order theory. *Composite Structures 89*, pages 1–3, 2009.

[38] D. Young. Vibration of rectangular plates by the Ritz method. *Journal of Applied Mechanics 17*, pages 448–453, 1950.

[39] A.F. Martin and A.W. Leissa. Application of the Ritz method to plane elasticity problems for composite sheets with variable fiber spacing. *International Journal for Numerical Methods in Engineering 28*, pages 1813–1825, 1989.

[40] R. Vescovini, L. Dozio, M. D'Ottavio, and O. Polit. On the application of the Ritz method to free vibration and buckling analysis of highly anisotropic plates. *Composite Structures 192*, pages 1–7, 2018.

[41] M. D'Ottavio, R. Vescovini, L. Dozio, and O. Polit. The Ritz-Sublaminate Generalized Unified Formulation approach for piezoelectric composite plates. *International Journal of Smart and Nano Materials 9*, pages 3–4, 2018.

[42] R. Bellman, B.G. Kashef, and J. Casti. Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *Journal of Computational Physics 10*, pages 40–52, 1972.

[43] C.Shu and H. Du. Implementation of clamped and simply supported boundary conditions in the General Differential Quadrature free vibration analysis of beams and plates. *International Journal of Solids and Structures 34*, pages 1–6, 1997.

[44] C. Shu and H.Du. A generalized approach for implementing general boundary conditions in the GDQ free vibration analysis of plates. *International Journal of Solids and Structures 34*, pages 1–10, 1996.

[45] F. Tornabene, N. Fantuzzi, and M. Bacciocchi. Strong and weak formulations based on differential and integral quadrature methods for the free vibration analysis of composite plates and shells: convergence and accuracy. *Engieering Analysis with Boundary Elements 92*, pages 1–4, 2018.

[46] F. Tornabene, N. Fantuzzi, and M. Bacciocchi. The local GDQ method for the natural frequencies of doubly-curved shells with variable thickness: A general formulation. *Composites Parts B 92*, pages 1–8, 2016.

[47] R. Boutaba, M.A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O.M. Caicedo. A comprehensive survey on Machine Learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications 9*, pages 11–15, 2018.

[48] E. Mjolsness and D. DeCoste. Machine Learning for science: State of the art and future prospects. *American Association for the Advancement of Science*, pages 2051–2055, 2001.

[49] O.I. Abiodun, A. Jantan, A.E. Omalara, K.V. Dada, N.A. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon 4*, pages 1–11, 2012.

[50] J.L. Rogers. Simulating structural analysis with neural network. *Journal of Computing in Civil Engineering 8*, pages 1–6, 1994.

[51] N.D. Lagaros and M. Papadrakakis. Learning improvement of neural networks used in structural optimization. *Advances in Engineering Software 35*, pages 1–2, 2003.

[52] L.C. Nguyen and H. Nguyen-Xuan. Deep learning for computational structural optimization. *ISA Transactions 103*, pages 1–4, 2020.

[53] Y. Lin, Z. Nie, and H. Ma. Structural damage detection with automatic feature extraction through Deep Learning. *Computer-Aided Civil and Infrastructure Engineering 32*, pages 1025–1046, 2017.

[54] S. Lee, J. Ha, M. Zokhirova, H. Moon, and J. Lee. Background information of Deep Learning for structural engineering. *Archives of Computational Methods in Engineering 25*, pages 121–129, 2018.

[55] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics 335*, pages 736–746, 2017.

[56] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Machine Learning of linear differential equations using Gaussian processes. *Journal of Computational Physics 348*, pages 683–693, 2017.

[57] M. Raissi and G.E. Karniadakis. Hidden physics models: Machine Learning of nonlinear partial differential equations. *Journal of Computational Physics 357*, pages 125–141, 2017.

[58] 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*, pages 2–10, 2018.

[59] E. Haghighat, A. Moure, M. Raissi, and H. Gómez. A deep learning framework for solution and discovery in solid mechanics: linear elasticity. pages 8–12, 2020.

[60] W. Li, M.Z. Bazant, and J. Zhu. A Physics-Guided Neural Network framework for elastic plates: comparison of governing equations-based and energy-based approaches. *Computer Methods in Applied Mechanics and Engineering 383*, pages 3–11, 2021.

[61] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, and T. Rabczuk. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementations and applications. *Computer Methods in Applied Mechanics and Engineering 362*, pages 1–29, 2020.

[62] G.B. Huang, Q.Y. Zhu, and C.K. Siew. Extreme Learning Machine: theory and applications. *Neurocomputing 70*, pages 1–5, 2005.

[63] G.B. Huang and H.A. Babri. Upper bounds on the number of hidden neurons in feed-forward networks with arbitrary bounded nonlinear activation functions. *IEEE Neural Networks 9*, pages 224–229, 1998.

[64] D. Xiao, B. Li, and Y. Mao. A Multiple Hidden Layers Extreme Learning Machine method and its application. *Mathematical Problems in Engineering 2017*, pages 3–5, 2017.

[65] *ASM Handbook Volume 2: Properties and Selection: Nonferrous Alloys and Special-Purpose Materials*. ASM International, 10th edition, 1990.

[66] N.J. Pagano. Exact solutions for composite laminates in cylindrical bending. *Journal of Composite Materials 3*, pages 8–13, 1969.

[67] K.T.S.R. Iyengar, K. Chandrashekhara, and V.K. Sebastian. On the analysis of thick rectangular plates. *Archive of Applied Mechanics 43*, pages 6–12, 1974.

[68] T. Kant. Numerical analysis of thick plates. *Computer Methods in Applied Mechanics and Engineering 31*, pages 6–9, 1982.

Appendices

A Expanded formulation in SGUF theory

The expanded form of the PVW in the context of multilayered plates regarding SGUF theory is reported in Eq.(64):

$$
\begin{aligned}
&\sum_{k=1}^{N_k}\sum_{p=1}^{N_p^k}\int_\Omega \{\delta u^{p,k}_{x\alpha_{ux}}[\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x} u^{p,k}_{x\beta_{ux},y} \\
&+ \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y} u^{p,k}_{y\beta_{uy},x}] + \delta u^{p,k}_{y\alpha_{uy},y}[\tilde{C}^{p,k}_{12} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y} u^{p,k}_{y\beta_{uy},y} \\
&+ \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x} u^{p,k}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y} u^{p,k}_{y\beta_{uy},x}] + \delta u^{p,k}_{x\alpha_{ux},y}[\tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x} u^{p,k}_{x\beta_{ux},x} \\
&+ \tilde{C}^{p,k}_{26} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x} u^{p,k}_{x\beta_{ux},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y} u^{p,k}_{y\beta_{uy},x}] \\
&+ \delta u^{p,k}_{y\alpha_{uy},x}[\tilde{C}^{p,k}_{16} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x} u^{p,k}_{x\beta_{ux},x} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y} u^{p,k}_{y\beta_{uy},y} + \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x} u^{p,k}_{x\beta_{ux},y} \\
&+ \tilde{C}^{p,k}_{66} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y} u^{p,k}_{y\beta_{uy},x}] + \delta u^{p,k}_{z\alpha_{uz},x}[\tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{uz}\beta_{ux}}_{u_z\partial u_x} u^{p,k}_{x\beta_{ux}} \\
&+ \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z} u^{p,k}_{z\beta_{uz},y} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uy}}_{u_z\partial u_y} u^{p,k}_{y\beta_{uy}}] + \delta u^{p,k}_{z\alpha_{uz},y}[\tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z} u^{p,k}_{z\beta_{uz},x} \\
&+ \tilde{C}^{p,k}_{45} Z^{p\alpha_{uz}\beta_{ux}}_{u_z\partial u_x} u^{p,k}_{x\beta_{ux}} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z} u^{p,k}_{z\beta_{uz},y} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uz}\beta_{uy}}_{u_z\partial u_y} u^{p,k}_{y\beta_{uy}}] \\
&+ \delta u^{p,k}_{x\alpha_{ux}}[\tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_xu_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{55} Z^{p\alpha_{ux}\beta_{ux}}_{\partial u_x\partial u_x} u^{p,k}_{x\beta_{ux}} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_xu_z} u^{p,k}_{z\beta_{uz},y} \\
&+ \tilde{C}^{p,k}_{45} Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_x\partial u_y} u^{p,k}_{y\beta_{uy}}] + \delta u^{p,k}_{y\alpha_{uy}}[\tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_y\partial u_z} u^{p,k}_{z\beta_{uz},x} + \tilde{C}^{p,k}_{45} Z^{p\alpha_{uy}\beta_{ux}}_{\partial u_y\partial u_x} u^{p,k}_{x\beta_{ux}} \\
&+ \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_yu_z} u^{p,k}_{z\beta_{uz},y} + \tilde{C}^{p,k}_{44} Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_y\partial u_y} u^{p,k}_{y\beta_{uy}}] + \delta u^{p,k}_{x\alpha_{ux},x}\tilde{C}^{p,k}_{13} Z^{p\alpha_{ux}\beta_{uz}}_{u_x\partial u_z} u^{p,k}_{z\beta_{uz}} \\
&+ \delta u^{p,k}_{y\alpha_{uy},y}\tilde{C}^{p,k}_{23} Z^{p\alpha_{ux}\beta_{uz}}_{u_y\partial u_z} u^{p,k}_{z\beta_{uz}} + \delta u^{p,k}_{x\alpha_{ux},y}\tilde{C}^{p,k}_{36} Z^{p\alpha_{ux}\beta_{uz}}_{u_x\partial u_z} u^{p,k}_{z\beta_{uz}} + \delta u^{p,k}_{y\alpha_{uy},x}\tilde{C}^{p,k}_{36} Z^{p\alpha_{uy}\beta_{uz}}_{u_y\partial u_z} u^{p,k}_{z\beta_{uz}} \\
&+ \delta u^{p,k}_{z\alpha_{uz}}\tilde{C}^{p,k}_{13} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_zu_x} u^{p,k}_{x\beta_{ux},x} + \delta u^{p,k}_{z\alpha_{uz}}\tilde{C}^{p,k}_{23} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_zu_y} u^{p,k}_{y\beta_{uy},y} + \delta u^{p,k}_{z\alpha_{uz}}\tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_zu_x} u^{p,k}_{x\beta_{ux},y} \\
&+ \delta u^{p,k}_{z\alpha_{uz}}\tilde{C}^{p,k}_{36} Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_zu_y} u^{p,k}_{y\beta_{uy},x} + \delta u^{p,k}_{z\alpha_{uz}}\tilde{C}^{p,k}_{33} Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z\partial u_z} u^{p,k}_{z\beta_{uz}}\}d\Omega + \\
&\sum_{k=1}^{N_k}\sum_{p=1}^{N_p^k}\int_\Omega [\delta u^{p,k}_{x\alpha_{ux}}\rho^{p,k} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x}\ddot{u}^{p,k}_{x\beta_{ux}} + \delta u^{p,k}_{y\alpha_{uy}}\rho^{p,k} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y}\ddot{u}^{p,k}_{y\beta_{uy}} \\
&+ \delta u^{p,k}_{z\alpha_{uz}}\rho^{p,k} Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z}\ddot{u}^{p,k}_{z\beta_{uz}}]d\Omega = \int_\Omega [\delta u^{N_p^k,N_k}_{z0} f_z^{top} + \delta u^{1,1}_{z1} f_z^{bot}]d\Omega
\end{aligned}
\tag{64}
$$

The expanded form of the PVW in the context of multilayered shells regarding SGUF theory is reported in Eq.(65):

$$
\begin{aligned}
&\sum_{k=1}^{N_k}\sum_{p=1}^{N_p^k}\int_\Omega \{\frac{\partial\delta u^{p,k}_{x\alpha_{ux}}}{\partial x}[\tilde{C}^{p,k}_{11} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x00}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial x} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y11}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y} + \tilde{C}^{p,k}_{12} Z^{p\alpha_{ux}\beta_{uz}}_{u_xu_z10} u^{p,k}_{z\beta_{uz}} \\
&+ \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x11}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y} + \tilde{C}^{p,k}_{16} Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y00}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x} + \tilde{C}^{p,k}_{13} Z^{p\alpha_{ux}\beta_{uz}}_{u_x\partial u_z00} u^{p,k}_{z\beta_{uz}}] \\
&+ \frac{\partial\delta u^{p,k}_{y\alpha_{uy}}}{\partial y}[\tilde{C}^{p,k}_{12} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x11}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial x} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y22}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y} + \tilde{C}^{p,k}_{22} Z^{p\alpha_{uy}\beta_{uz}}_{u_yu_z21} u^{p,k}_{z\beta_{uz}} \\
&+ \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x22}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y} + \tilde{C}^{p,k}_{26} Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y11}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x} + \tilde{C}^{p,k}_{23} Z^{p\alpha_{uy}\beta_{uz}}_{u_y\partial u_z11} u^{p,k}_{z\beta_{uz}}]
\end{aligned}
$$

$$
\begin{aligned}
&+\partial\delta u^{p,k}_{z\alpha_{uz}}[\tilde{C}^{p,k}_{12}Z^{p\alpha_{uz}\beta_{ux}}_{u_zu_x10}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial x}+\tilde{C}^{p,k}_{22}Z^{p\alpha_{uz}\beta_{uy}}_{u_zu_y21}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y}+\tilde{C}^{p,k}_{22}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z20}u^{p,k}_{z\beta_{uz}}\\
&+\tilde{C}^{p,k}_{26}Z^{p\alpha_{uz}\beta_{ux}}_{u_zu_x21}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y}+\tilde{C}^{p,k}_{26}Z^{p\alpha_{uz}\beta_{uy}}_{u_zu_y10}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x}+\tilde{C}^{p,k}_{23}Z^{p\alpha_{uz}\beta_{uz}}_{u_z\partial u_z10}u^{p,k}_{z\beta_{uz}}]\\
&+\frac{\partial\delta u^{p,k}_{x\alpha_{ux}}}{\partial y}[\tilde{C}^{p,k}_{16}Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x11}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial x}+\tilde{C}^{p,k}_{26}Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y22}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y}+\tilde{C}^{p,k}_{26}Z^{p\alpha_{ux}\beta_{uz}}_{u_xu_z21}u^{p,k}_{z\beta_{uz}}\\
&+\tilde{C}^{p,k}_{66}Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x22}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y}+\tilde{C}^{p,k}_{66}Z^{p\alpha_{ux}\beta_{uy}}_{u_xu_y11}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x}+\tilde{C}^{p,k}_{36}Z^{p\alpha_{ux}\beta_{uz}}_{u_x\partial u_z11}u^{p,k}_{z\beta_{uz}}]\\
&+\frac{\partial\delta u^{p,k}_{y\alpha_{uy}}}{\partial x}[\tilde{C}^{p,k}_{16}Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x00}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial x}+\tilde{C}^{p,k}_{26}Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y11}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y}+\tilde{C}^{p,k}_{26}Z^{p\alpha_{uy}\beta_{uz}}_{u_yu_z10}u^{p,k}_{z\beta_{uz}}\\
&+\tilde{C}^{p,k}_{66}Z^{p\alpha_{uy}\beta_{ux}}_{u_yu_x11}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y}+\tilde{C}^{p,k}_{66}Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y00}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x}+\tilde{C}^{p,k}_{36}Z^{p\alpha_{uy}\beta_{uz}}_{u_y\partial u_z00}u^{p,k}_{z\beta_{uz}}]\\
&+\partial\delta u^{p,k}_{y\alpha_{uy}}[\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y20}\partial u^{p,k}_{y\beta_{uy}}-\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uy}}_{u_y\partial u_y10}u^{p,k}_{y\beta_{uy}}-\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uz}}_{u_yu_z21}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial y}\\
&-\tilde{C}^{p,k}_{45}Z^{p\alpha_{uy}\beta_{ux}}_{u_y\partial u_x10}u^{p,k}_{x\beta_{ux}}-\tilde{C}^{p,k}_{45}Z^{p\alpha_{uy}\beta_{uz}}_{u_yu_z10}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial x}]\\
&+\partial\delta u^{p,k}_{y\alpha_{uy}}[-\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_yu_y10}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uy}}_{\partial u_y\partial u_y00}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{44}Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_yu_z11}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial y}\\
&+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uy}\beta_{ux}}_{\partial u_y\partial u_x00}u^{p,k}_{x\beta_{ux}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uy}\beta_{uz}}_{\partial u_yu_z00}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial x}]\\
&+\frac{\partial\delta u^{p,k}_{z\alpha_{uz}}}{\partial y}[-\tilde{C}^{p,k}_{44}Z^{p\alpha_{uz}\beta_{uy}}_{u_zu_y21}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{44}Z^{p\alpha_{uz}\beta_{uy}}_{u_z\partial u_y11}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{44}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z22}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial y}\\
&+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uz}\beta_{ux}}_{u_z\partial u_x11}u^{p,k}_{x\beta_{ux}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z11}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial x}]\\
&+\delta u^{p,k}_{x\alpha_{ux}}[-\tilde{C}^{p,k}_{45}Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_xu_y10}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{ux}\beta_{uy}}_{\partial u_x\partial u_y00}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_xu_z11}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial y}\\
&+\tilde{C}^{p,k}_{55}Z^{p\alpha_{ux}\beta_{ux}}_{\partial u_x\partial u_x00}u^{p,k}_{x\beta_{ux}}+\tilde{C}^{p,k}_{55}Z^{p\alpha_{ux}\beta_{uz}}_{\partial u_xu_z00}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial x}]\\
&+\frac{\partial\delta u^{p,k}_{z\alpha_{uz}}}{\partial x}[-\tilde{C}^{p,k}_{45}Z^{p\alpha_{uz}\beta_{uy}}_{u_zu_y10}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uz}\beta_{uy}}_{u_z\partial u_y00}u^{p,k}_{y\beta_{uy}}+\tilde{C}^{p,k}_{45}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z11}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial y}\\
&+\tilde{C}^{p,k}_{55}Z^{p\alpha_{uz}\beta_{ux}}_{u_z\partial u_x00}u^{p,k}_{x\beta_{ux}}+\tilde{C}^{p,k}_{55}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z00}\frac{\partial u^{p,k}_{z\beta_{uz}}}{\partial x}]\\
&+\delta u^{p,k}_{z\alpha_{uz}}[\tilde{C}^{p,k}_{13}Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_zu_x00}\frac{\partial u^{p,k}_{x\beta_{ux}}}{x}+\tilde{C}^{p,k}_{23}Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_zu_y11}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial y}+\tilde{C}^{p,k}_{23}Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_zu_z10}u^{p,k}_{z\beta_{uz}}\\
&+\tilde{C}^{p,k}_{36}Z^{p\alpha_{uz}\beta_{ux}}_{\partial u_zu_x11}\frac{\partial u^{p,k}_{x\beta_{ux}}}{\partial y}+\tilde{C}^{p,k}_{36}Z^{p\alpha_{uz}\beta_{uy}}_{\partial u_zu_y00}\frac{\partial u^{p,k}_{y\beta_{uy}}}{\partial x}+\tilde{C}^{p,k}_{33}Z^{p\alpha_{uz}\beta_{uz}}_{\partial u_z\partial u_z00}u^{p,k}_{z\beta_{uz}}]\}d\Omega\\
&\sum_{k=1}^{N_k}\sum_{p=1}^{N^k_p}\int_\Omega[\delta u^{p,k}_{x\alpha_{ux}}\rho^{p,k}Z^{p\alpha_{ux}\beta_{ux}}_{u_xu_x00}\ddot{u}^{p,k}_{x\beta_{ux}}+\delta u^{p,k}_{y\alpha_{uy}}\rho^{p,k}Z^{p\alpha_{uy}\beta_{uy}}_{u_yu_y00}\ddot{u}^{p,k}_{y\beta_{uy}}\\
&+\delta u^{p,k}_{z\alpha_{uz}}\rho^{p,k}Z^{p\alpha_{uz}\beta_{uz}}_{u_zu_z00}\ddot{u}^{p,k}_{z\beta_{uz}}]d\Omega=\int_\Omega[\delta u^{N^k_p,N_k}_{z0}f^{top}_z(1+\frac{h}{2R})+\delta u^{1,1}_{z1}f^{bot}_z(1-\frac{h}{2R})]d\Omega \qquad (65)
\end{aligned}
$$

Partitions of the constitutive matrix in the context of SGUF theory are used in the definition

of in-plane and transverse components of the constitutive relation. They are given in Eq.(66):

$$\tilde{\boldsymbol{C}}^{p,k}_{\Omega\Omega} = \begin{bmatrix} \tilde{C}^{p,k}_{11} & \tilde{C}^{p,k}_{12} & \tilde{C}^{p,k}_{16} \\ \tilde{C}^{p,k}_{12} & \tilde{C}^{p,k}_{22} & \tilde{C}^{p,k}_{26} \\ \tilde{C}^{p,k}_{16} & \tilde{C}^{p,k}_{26} & \tilde{C}^{p,k}_{66} \end{bmatrix} \qquad \tilde{\boldsymbol{C}}^{p,k}_{\Omega n} = \begin{bmatrix} 0 & 0 & \tilde{C}^{p,k}_{13} \\ 0 & 0 & \tilde{C}^{p,k}_{23} \\ 0 & 0 & \tilde{C}^{p,k}_{36} \end{bmatrix}$$

$$\tilde{\boldsymbol{C}}^{p,k}_{n\Omega} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{C}^{p,k}_{13} & \tilde{C}^{p,k}_{23} & \tilde{C}^{p,k}_{36} \end{bmatrix} \qquad \tilde{\boldsymbol{C}}^{p,k}_{nn} = \begin{bmatrix} \tilde{C}^{p,k}_{44} & \tilde{C}^{p,k}_{45} & 0 \\ \tilde{C}^{p,k}_{45} & \tilde{C}^{p,k}_{55} & 0 \\ 0 & 0 & \tilde{C}^{p,k}_{33} \end{bmatrix} \tag{66}$$

B Additional results

Layup	Boundary conditions	x=	y=	Ritz [m]	Present [m]
$[30, 60, 90]^\circ$ (5)	CCCC	$a/4$	$b/4$	-3.79 x 10^{-5}	-3.78 x 10^{-5}
		$-a/4$	$-b/4$	-3.79 x 10^{-5}	-3.78 x 10^{-5}
		$a/2$	$b/2$	-6.93 x 10^{-5}	-6.92 x 10^{-5}
	SSSS	$a/4$	$b/4$	-1.69 x 10^{-4}	-9.14 x 10^{-5}
		$-a/4$	$-b/4$	-1.69 x 10^{-4}	-9.14 x 10^{-5}
		$a/2$	$b/2$	-2.72 x 10^{-4}	-2.24 x 10^{-4}
	CCSS	$a/4$	$b/4$	-9.59 x 10^{-5}	-8.62 x 10^{-5}
		$-a/4$	$-b/4$	-6.10 x 10^{-5}	-4.91 x 10^{-5}
		$a/2$	$b/2$	-1.30 x 10^{-4}	-1.24 x 10^{-4}
$[82, -74, 88]^\circ$ (4)	CCCC	$a/4$	$b/4$	-1.34 x 10^{-5}	-1.34 x 10^{-5}
		$-a/4$	$-b/4$	-1.34 x 10^{-5}	-1.34 x 10^{-5}
		$a/2$	$b/2$	-2.37 x 10^{-5}	-2.37 x 10^{-5}
	SSSS	$a/4$	$b/4$	-8.28 x 10^{-5}	-8.17 x 10^{-5}
		$-a/4$	$-b/4$	-8.28 x 10^{-5}	-8.17 x 10^{-5}
		$a/2$	$b/2$	-1.16 x 10^{-4}	-1.19 x 10^{-4}
	CCSS	$a/4$	$b/4$	-3.99 x 10^{-5}	-4.01 x 10^{-5}
		$-a/4$	$-b/4$	-2.23 x 10^{-5}	-2.28 x 10^{-5}
		$a/2$	$b/2$	-4.70 x 10^{-5}	-4.72 x 10^{-5}
$[-45, 7, 85]^\circ$ (6)	CCCC	$a/4$	$b/4$	-3.82 x 10^{-5}	-3.80 x 10^{-5}
		$-a/4$	$-b/4$	-3.82 x 10^{-5}	-3.80 x 10^{-5}
		$a/2$	$b/2$	-7.17 x 10^{-5}	-7.17 x 10^{-5}
	SSSS	$a/4$	$b/4$	-1.73 x 10^{-4}	-1.53 x 10^{-4}
		$-a/4$	$-b/4$	-1.73 x 10^{-4}	-1.53 x 10^{-4}
		$a/2$	$b/2$	-2.89 x 10^{-4}	-2.88 x 10^{-4}
	CCSS	$a/4$	$b/4$	-9.29 x 10^{-5}	-8.98 x 10^{-5}
		$-a/4$	$-b/4$	-6.11 x 10^{-5}	-5.00 x 10^{-5}
		$a/2$	$b/2$	-1.32 x 10^{-4}	-1.29 x 10^{-4}

Table 27: Layup sequences and composite materials for different boundary conditions. Additional results for Table [8]. Transverse displacement evaluated at midsurface. Case setup from *Configuration 2*.

Aspect ratio (AR)	Skew angle (γ)	Value	x=	y=	z=	Ritz	Present	FEM
1.25	15° ($\lambda = 3 * 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	9.89 x 10^{-5}	9.69 x 10^{-5}	9.88 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0786	0.155	-0.07979
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.110	0.197	0.114
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0307	0.0374	0.0292
	30° ($\lambda = 2 * 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	6.72 x 10^{-5}	6.88 x 10^{-5}	6.72 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0578	-0.110	-0.0619
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0823	0.153	0.0959
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0225	0.0231	0.0238
	45° ($\lambda = 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	3.24 x 10^{-5}	3.32 x 10^{-5}	3.24 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0323	-0.0633	-0.0385
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0478	0.100	0.0684
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0114	0.0129	0.0141
	60° ($\lambda = 10^{10}$)	$u_z[m]$	$a/2$	$b/2$	0	8.99 x 10^{-6}	8.86 x 10^{-6}	8.98 x 10^{-6}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0113	-0.0265	-0.0169
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0186	0.0487	0.0372
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	3094.6	3720.6	5103.7
	75° ($\lambda = 9 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	7.08 x 10^{-7}	7.24 x 10^{-7}	7.06 x 10^{-7}
		σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	-1220	-6447.1	-3567.3
		σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	2840.8	14342	10636
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	278.66	746.26	612.4
1.5	15° ($\lambda = 9 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	5.72 x 10^{-5}	5.82 x 10^{-5}	5.72 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0487	-0.108	-0.0496
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0893	0.1417	0.0924
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0210	0.0267	0.0215
	30° ($\lambda = 8 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	3.83 x 10^{-5}	3.74 x 10^{-5}	3.83 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0349	-0.0746	-0.0379
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0663	0.127	0.0764
		τ_{xy} [MPa]	$a/4$	$b/4$	$-h/2$	0.0158	0.0175	0.0167
	45° ($\lambda = 7 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	1.80 x 10^{-5}	1.77 x 10^{-5}	1.80 x 10^{-5}
		σ_{xx} [MPa]	$a/2$	$b/2$	$-h/2$	-0.0183	-0.0439	-0.0229
		σ_{yy} [MPa]	$a/2$	$b/2$	$h/2$	0.0378	0.0746	0.0531
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	8518.2	9816.1	9949.5
	60° ($\lambda = 8 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	4.73 x 10^{-6}	4.72 x 10^{-6}	4.73 x 10^{-6}
		σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	-5709.9	-18676	-9783.8
		σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	13980	35901	27581
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	2598.5	3730.1	3432.1
	75° ($\lambda = 7 * 10^{9}$)	$u_z[m]$	$a/2$	$b/2$	0	3.46 x 10^{-7}	3.38 x 10^{-7}	3.45 x 10^{-7}
		σ_{xx} [Pa]	$a/2$	$b/2$	$-h/2$	-579.61	-4404.2	-2236.9
		σ_{yy} [Pa]	$a/2$	$b/2$	$h/2$	1912.6	9229.1	7427.3
		τ_{xy} [Pa]	$a/4$	$b/4$	$-h/2$	255.34	423.60	490.5

Table 28: Displacement and stress results comparison for the cases of skew plates from [1]. Additional results for Table [17].

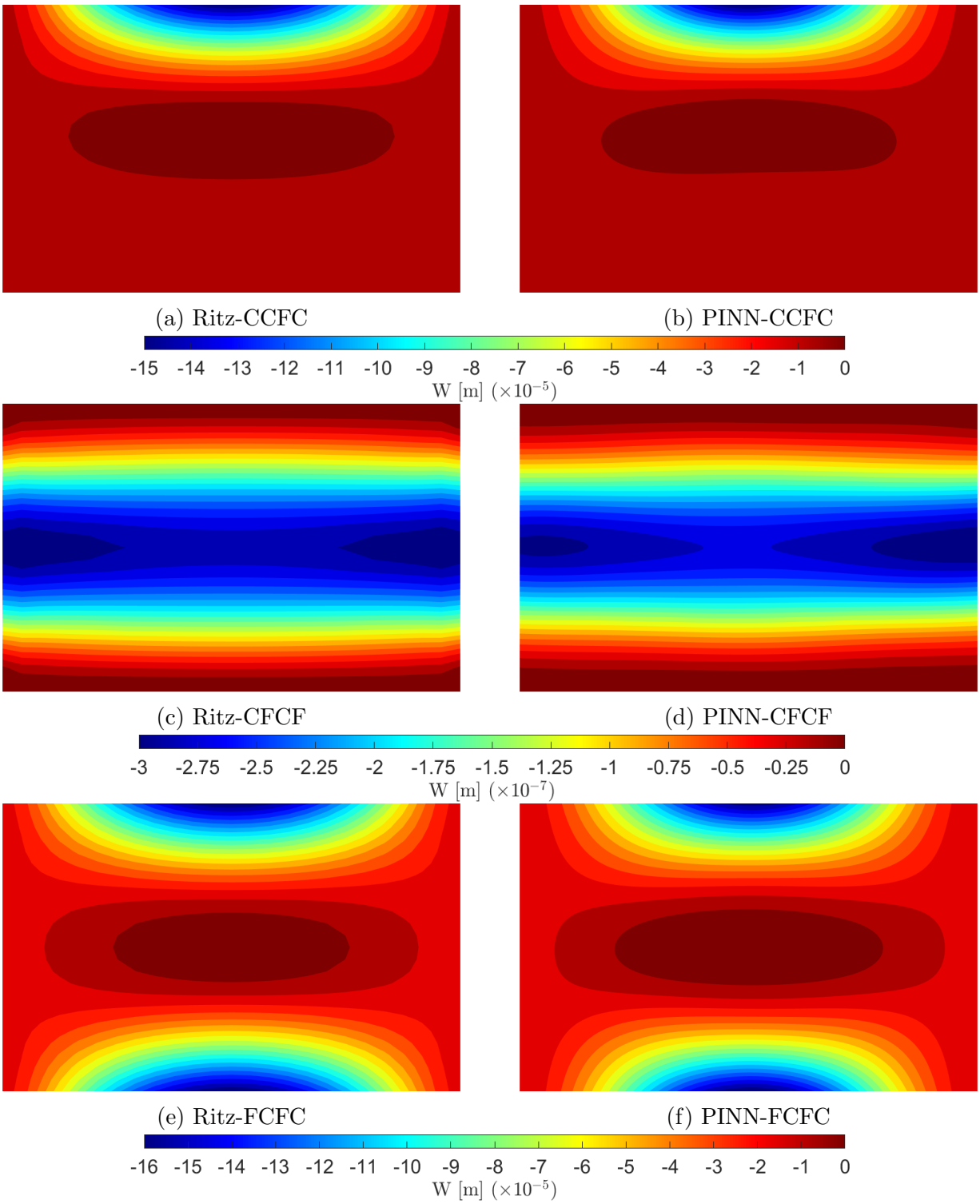
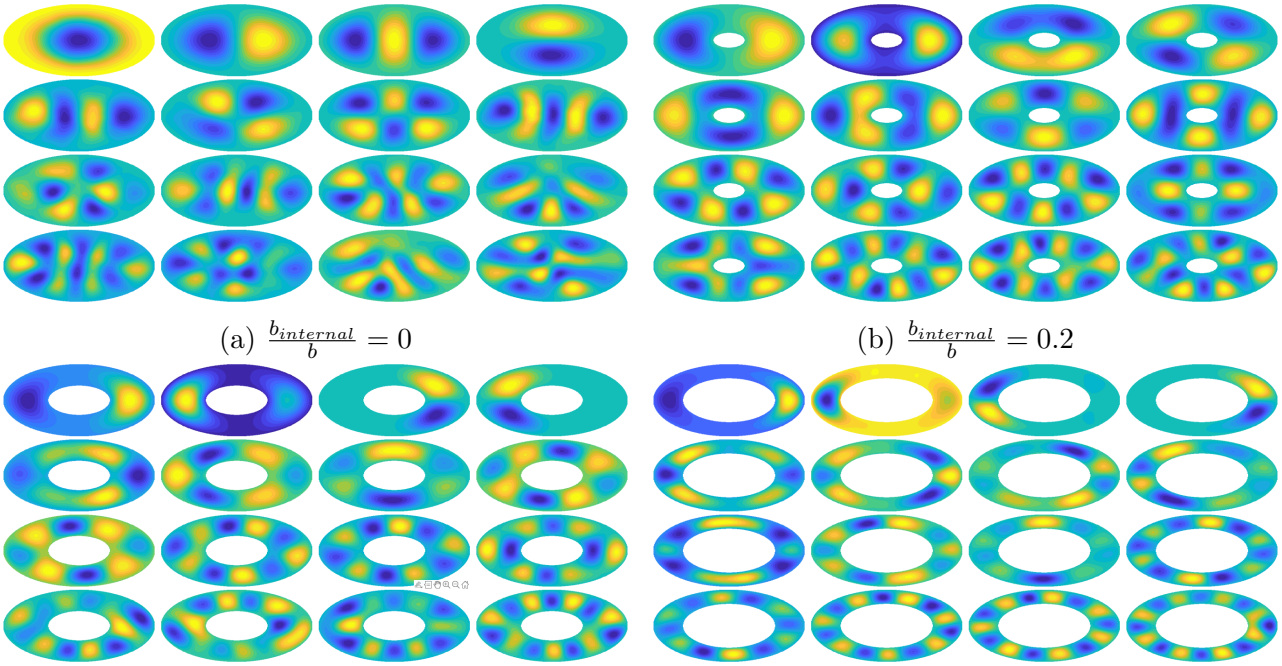


Figure 46: Contours of transverse displacement of shell configuration: $R = 100mm$, $\phi = \pi/2$, $h = 1mm$ from Table [16]. Boundary conditions: CCFC, CFCF and FCFC. Case setup from *Configuration 7*.



(a) $\frac{b_{internal}}{b} = 0$ (b) $\frac{b_{internal}}{b} = 0.2$

(c) $\frac{b_{internal}}{b} = 0.4$ (d) $\frac{b_{internal}}{b} = 0.6$

Figure 47: Contours of transverse displacements of the first 16 modes for elliptic plates in [2] with internal drilled section. Hole configurations: a)$\frac{b_{internal}}{b} = 0$, b)$\frac{b_{internal}}{b} = 0.2$, c)$\frac{b_{internal}}{b} = 0.4$ and d)$\frac{b_{internal}}{b} = 0.6$