Adaptation et hybridation du maillage pour une prédiction efficace des contraintes dans un modèle de plaque par couches

Lucille Salha

▶ **To cite this version:**

Lucille Salha. Adaptation et hybridation du maillage pour une prédiction efficace des contraintes dans un modèle de plaque par couches. Matériaux. Université Paris-Est; Université Saint-Joseph (Beyrouth). Faculté des Sciences, 2021. Français. NNT : 2021PESC1101 . tel-03587334

HAL Id: tel-03587334

https://pastel.hal.science/tel-03587334v1

Submitted on 24 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-EST

USJ 1875 Université Saint-Joseph de Beyrouth

Mesh adaptation and hybridization for efficient stress prediction in a layerwise plate model

THÈSE

présentée et soutenue publiquement le Date

pour l'obtention du grade du

Docteur de l'Université Paris-Est
et
Docteur de l'Université Saint-Joseph - Liban
(Spécialité Mathématiques)

par

Lucille SALHA

Composition du jury

Rapporteurs :	MARGUET Steven (HDR, Maitre de conférences)
	POLIT Olivier (HDR, Professeur)
Examinateurs :	BLEYER Jérémy (Chercheur)
	SAYAH Toni (HDR, Professeur)
Directeur de Thèse :	SAB Karam (HDR, Directeur de recherche)
Co-Encadrante de Thèse :	BODGI Joanna (Professeur assistant)

Laboratoire Navier (ENPC, Université Gustave Eiffel, CNRS UMR 8205) **Faculté des sciences**

Navier

Faculté des sciences

Acknowledgement

First of all, I would like to warmly thank my thesis supervisor, Professor Karam SAB, who integrated me into his team and gave me his trust. I send him my gratitude for the time granted, his kindness, his high standards, his scientific rigor, and the many encouragements lavished.

I would like to express my deep gratitude to Doctor Jérémy BLEYER for his supervision, his wise advice, his enthusiasm, his availability, and his involvement throughout this work. I really enjoyed working with him.

A big thank you to my thesis co-supervisor, Doctor Joanna BODGI for her supervision, her attention, and her advice which allowed me to carry out this work.
I would like to thank Sophia Vorderwuelbecke for her help in accessing Firedrake's latest development branches.
I would like to cordially thank Professor Olivier POLIT and Professor Steven MARGUET, for agreeing to be rapporteurs for this work.
My sincere thanks also go to Professor Tony SAYAH who accepted to chair my thesis jury.
I would like to express here all the pleasure I had working in the Navier laboratory. May all its members be thanked, in particular Marie-Françoise KASPI for her remarkable kindness and affection.
I also thank the doctoral students in France and Lebanon who worked with me during my thesis and with whom I shared unforgettable moments.

My thanks also go to my relatives and friends who, with this frequent question, "when are you going to defend your thesis ?" Though stressful, have been by my side to cheer me up all these years.

I dedicate this thesis to my husband and my family without whom I would not be where I am today. Their unconditional support and presence in difficult times have been indispensable to me.

Résumé

Les plaques multicouches sont devenues de plus en plus utiles dans le domaine de l'ingénierie, d'abord dans l'industrie, et plus récemment dans le génie civil. Qu'il s'agisse d'un mélange complexe de polymères, de bois ou de béton, des efforts importants sont nécessaires pour une modélisation précise de ces matériaux. En effet, des phénomènes induits d'anisotropie et d'hétérogénéité sont associés à ces multi-matériaux : effets de bord, dilatation thermique différentielle, délaminage/décollement ou non-linéarités de type viscosité endommagement, plasticité en couches ou interfaces. De nombreux modèles ont été développés ces dernières années dans le laboratoire Navier pour permettre une description suffisamment détaillée afin de répondre aux problèmes spécifiques mentionnés ci-dessus. En introduisant les contraintes d'interface comme des contraintes généralisées du modèle, ces approches ont démontré leur efficacité en ce qui concerne la représentation du champ de contrainte tridimensionnel. Ceci a permis de proposer des critères de d'endommagement et de modéliser de manière efficace le délaminage ou le décollement, phénomène très présent dans les composites multicouches. Dans cette étude nous commencerons donc par rappeler le modèle des contraintes statiquement compatible avec des approximations de contraintes de membrane de premier ordre par couche dans la direction de l'épaisseur (SCLS1), qui est un modèle par couches pour plaques multicouches. Le modèle SCLS1 est conforme exactement aux équations d'équilibre 3D et aux conditions aux limites de bord libre. En outre, une discrétisation par éléments finis basée sur le déplacement généralisé dual est mise en œuvre à l'aide du progiciel FEniCS et une stratégie de remaillage, basée sur un nouvel indicateur d'erreur, est proposée . Cet indicateur d'erreur est construit sur la base de la comparaison du champ de contraintes 3D directement déduit des contraintes généralisées par couches et d'un champ de contraintes reconstruit basé sur les déplacements généralisés du modèle. Comme il est difficile de construire des éléments finis conformes à base de contraintes assurant la continuité de la traction, nous considérons une stratégie d'hybridation dans laquelle cette condition est relaxée par l'introduction d'un multiplicateur de Lagrange supplémentaire de type déplacement, défini sur les facettes de l'élément. Par conséquent, cette méthode peut être utilisée pour capturer efficacement de fortes variations de contraintes intra- et inter-laminaires près des bords libres ou des fissures.
Les mots clé : stratifiés, modèle de plaque par couches, éléments finis mixtes, hybridation, bords libres.

Abstract

In engineering, industry, and, more recently, civil engineering, multi-layered plate are more useful today. Important efforts to model such materials accurately : a complex combination of polymers, wood, or concrete are necessary. This multi-material is in actual fact correlated with phenomena of induced anisotropy and heterogeneity : edge effects, differential thermal expansion, viscosity-style degradation or non linearities, layer or interface plasticity. In recent years in Navier Laboratory, several models were developed which permit sufficiently detailed descriptions to address the above specific issues. These approaches showed their effectiveness in depicting detail at inter-and intra-layer levels by incorporating interface stresses as generalized stresses in the model. It is then simple to provide damage laws to model delamination or debonding effectively, a very common phenomenon in multilayered composites. A multilayered plates layer model, labeled Statically Compatible Layerwise stresses with first-order membrane stress approaches per layer in the thickness direction (SCLS1), is therefore recalled in this analysis. The SCLS1 model complies exactly with the 3D equilibrium equations and the free-edge boundary conditions. Also, a dual displacement-based finite-element discretization is implemented using the FEniCS software package and a remeshing strategy is proposed based on a novel error indicator. The error indicator is built based on the comparison of the 3D stress field directly deduced from the layerwise generalized stresses and a reconstructed stress field based on the model generalized displacements.
Since conforming stress-based finite-elements ensuring traction continuity are difficult to construct, we consider a hybridization strategy in which traction continuity is relaxed by the introduction of an additional displacement such as the Lagrange multiplier defined on the element facets. As a result, this method can be used to capture efficiently strong intra- and inter-laminar stress variations near free-edges or cracks.
Keywords : laminates, layerwise plate model, Mixed finite element, Hybridization, free edges.

Table of contents

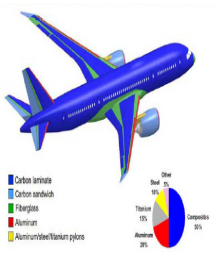
List of figures

List of tables

Introduction and bibliographical analysis

1 General introduction

In a process of continuous improvement, companies are developing new technological solutions to meet the expectations of their customers or even current economic and environmental constraints. These solutions go through the development and use of innovative materials in the products sold. For example, in the field of transport, the weight reduction of vehicles is at the heart of current questions on energy saving. To respond to this, the choice of materials has turned to multilayers, especially those made of composites, allowing lightness and strength to be combined. Their use has become widespread in other fields, such as sports in order to improve the performance of athletes, or even in the instrumentation sector to reduce the cost of manufacturing instruments. Multilayers are also used in the building industry where sound insulation is an important criterion. Conventional materials (concrete walls for example) have a sound reduction index proportional to their thickness. The use of thick walls is however impossible to implement from an economic point of view. Another form of multilayer material, called a sandwich, was then adopted in order to achieve performance similar or superior to conventional materials for lower wall thicknesses and reduced cost. Sandwiches are also used in the field of transportation (automotive glazing, for example) or personal protection (helmets, bulletproof vests). Figure 1 shows some applications involving multilayer materials in industry.



(a) Different types of materials used in Boeing 787

(b) Dassi brand composite bicycle frame

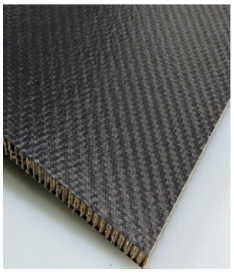
(c) Roof made with sandwich material

FIGURE 1 – Examples of applications of multilayer materials

The term composite encompasses all materials combining a matrix (resin, elastomer, rubber, cement) with reinforcement (fabrics, sand, carbon fibers, wood, glass). The geometric distribution of these matrices and reinforcements as well as their interactions make it possible to create different varieties of composites. The reinforcements modify the mechanical properties of the matrix by creating an orthotropy. In the case of a multilayer, the orientation of the fibers of each of the layers makes it possible to adjust the mechanical properties according to the required needs. Sandwiches, on the other hand, are multilayered structures whose layers, glued together, have different and complementary mechanical properties. They generally combine stiff materials (metals, wood, composites) with flexible and damping materials (honeycombs, porous, polystyrene). Examples of multilayers are shown in Figure 2.

(a) Carbon fiber composite plates

(b) Wooden sandwich structure with a polystyrene core



(c) Sandwich structure in a honeycomb with composite skins

FIGURE 2 – Different varieties of multilayers

Multilayer plates are used in different branches of technology. In fact, they have very interesting thermal properties and they can be rigid even for low densities. However, studies have shown problematic behaviours of multilayered plates near free edges, because of the differences in adjacent layers properties. Thus, the modelisation of these plates need a development of mathematical and numerical tools, that take into account their specific properties. In terms of memory space and computer time, the 3D models are so costly. Many models are already proposed but they have proved ineffective in modeling the behaviour of multilayered plates near free edges. In view of their structures, multilayered plates can be presented as a superposition of many anisotropic and homogeneous layers. Multilayer materials are ultimately appreciated for their modular mechanical properties that conventional homogeneous materials cannot offer. Knowledge of their mechanical behavior is however necessary to be able to fully exploit their potential. The design process is based on the modeling of the structures and the material parameters.

However, modeling these composites using finite element (FE) three-dimensional models is expensive in terms of computational time and memory. Consequently, this method of calculation can be used in specific regions of the structures and give reference results for specific configurations of the plate. The drawbacks of non-convergent results close to the edge and interfaces between layers due to the existence of singularities were shown by (Leguillon and E.Sanchez-Palencia (1987) ;Ting and Chou (1981) ; Wang and Choi (1982) ; Leguillon (1999) ; Chue and Liu (2002) ;Mittelsteda and Becker (2005)). In addition, the differences of elastic properties between layers result into a concentrated interlaminar stress near free edges. The mesh should be refined in order to capture the stresses, delamination and damage. For these purposes, the existence of special models for composite materials has proved to be important in view of the relatively limited thickness of multilayered structures.

Many 2D models have been proposed in order to model the 3D composite structures and take into account the effects between the interfaces of layers. The theories are grouped in two categories, the mono-layer equivalent models and the discrete layer models.

The first category includes the Kirchhoff-Love models, Reissner-Mindlin models (First order shear deformation models) and the higher order shear deformation. The theories based on Kirchhoff-Love models, eliminate the transverse shear stresses (Yang et al. (1996) ; Reissner and Stavsky (1961)), and are only valid for thin plates. Using Kirchhoff-Love theory for thick multilayered plates with anisotropy leads to numerical errors and gives wrong results. The role of shear transvese is more important in anisotropic structures then in isotropic structures.

The Reissner-Mindlin theory (Mindlin (1951)) can model thick plates and take into considertaion the shear transverse stresses. Theory of first order postulates first degree kinematics in co-ordinate thickness z (Whitney and Pagano (1970) ; Reissner (1972)). Several advances were made with the so-called first order theory of shear deformation, Daghia et al. (2008).

Theories of the higher order were based on nonlinear 3D displacement, stress approximations in z or mixed approximation (Tarun et al. (1982) ; Reissner (1984) ; Reddy (1984)). They are applicable to the multilayered and composite laminates. Based on Reissner-Mindlin theory, a Bending-Gradient model was developed in Navier Laboratory (Lebée and Sab (2011a), 2011b).

In addition, (Di Sciuva (1984) ; Di Sciuva (1986) ; Marco and Ugo (1993)) proposed a "zig-zag" model, that takes into account the displacement fields distributions specially at the interface between layers.

In the second group of models, the models are based on the layered approach. The fields are linear and quadratic across each thickness of the layer (Di Sciuva (1984) ; Pagano and Soni (1983) ; Carrera (2000), 2002, 1998, 1999a, 1999b). These models are advanced models that allow local responses to be investigated, particularly on the layer interface. Moreover, the number of variables and equations in those models depends on the number of layers in the plate which increases the calculation amount.
With direct influence from Pagano's model (Pagano (1978)), a layerwise stress model was presented in (Chabot (1997) ; Carreria et al. (2002) ; Diaz et al. (2002) ; Nguyen and Caron (2006) ;Dallot and Sab (2008) ;Saeedi et al. (2012a), 2012b, 2013a, 2013b ;Lerpiniere et al. (2014)). The multi-layered material in this model is seen as an overlay of Reissner-Mindlin plates connected by interfacial stresses which are seen as additional generalized stresses. This model firstly named Multi-particle model for Multilayered plates (M4), was studied by Chabot (1997). In Chabot (1997), Chabot presented many M4 models starting from M4-7n approach similar to Pagano model till she arrived to Kirchhoff-Love theory. Many models of M4 family were proposed by (Tran et al. (2004) ; Chabot et al. (2007) ; Chabot et al. (2013) ; Nasser et al. (2016) ; Nasser and Chabot (2018) ; Chabot and Deep (2019) ; Nasser et al. (2018a)) and prove their accuracy near free edges and at the interfaces bet-

ween layers by comparison to the 3D finite element calculation.

Adopting Carrera's proposed nomenclature (Carrera (2004)) the M4-5n model was renamed as LS1, the first order membrane stress approximations by layer in the thickness direction.

One of the key distinctions between the LS1 model and other current layer-wise models is that other models have either a mixed approach or displacement approach, while the LS1 model is only stress-based approach where there are no assumptions on displacement fields. Although the LS1 is very successful with its refined version, they are still possible improvement. First, such model cannot fulfill the 3D stress-free boundary conditions. Secondly, as the number of mathematical layers per physical layer increases there is no guarantee that the refined LS1 model converges to the 3D model.

To this purpose, an enhanced model proposed by Baroud et al. (2016) known as Statically Compatible LS1 (SCLS1) or (M4(6n-1)) is derived from the minimum potential complementary energy theorem which ensures that its refined version converges to the exact 3D model as the number of mathematic layers increases.

The object of this study is to find a mesh adaptation strategy for the SCLS1 model that allows further developments including interfaces delamination propagation or damage at the ply level. Then, we aim to make further improvements on the model using a hybrid mixed approach.
This dissertation consists of four chapters : First, a state-of-the-art approach which refers to the different approaches and advantages of existing multilayered plates.
The SCLS1 layerwise model is recalled in the second chapter. The Statically Compatible Layerwise stresses with first-order stress approximations per layer (SCLS1) model complies with 3D equilibrium equations and with free-edge boundary conditions. The laminated plate is seen as an overlay of Reissner plates, combined with stresses on the interfaces. In particular, the divergence of the interlaminar transverse shears are incorporated in the SCLS1 model as additional generalized stress. This model is based on the minimum of the complementary energy theorem as opposed to the LS1 model, which stems from Hellinger-Reissner. This ensures that the refined SCLS1 solution converges to the exact 3D solution with an increasing number of layers per physical layer.

In the third chapter, a mesh adaptation strategy is developed which relies on the reconstruction of 3D displacement fields from the SCLS1 model generalized displacements. Some illustrative examples show that the method is in fact able to refine the mesh in regions with complex 3D stress fields such as straight edges, notches or delamination fronts.

In the last chapter, an alternative finite element discretization to the conventional displacement-based finite-element method will be examined. Then, we will consider a hybridization strategy in which traction continuity is modified by the introduction of an additional displacement-like Lagrange multiplier defined on the elements facets. It will be shown on

some examples that the proposed mixed hybridized mixed approach is more accurate than a standard displacement approach for the same number of degrees of freedom.

Finally, we propose some conclusions and perspectives for this work.

2 Bibliographical analysis

Models of multilayered structures in the litterature

Plates are flat structures of largely small thickness used in industry and in civil engineering construction, such as car bodies, gas or liquid tanks, airplane wings. During the last decade, composite laminates have been widely used in different branches of technology. In fact, they have very interesting thermal properties and they can be rigid even for low densities. The multi-layered plates are being increasingly used in the aeronautical and aerospace industry. Multilayered plates, are plates formed of superposed layers where each layer have its own properties such as materials and thickness. So, multilayered plates are considered as a heterogeneous body consisting of homogeneous layers bounded together. The shaping of a modern, multi-layered structure with a strong anisotropy requires refined theory, with a good transverse shears and interlaminar stress definition in mind.
Over several years, two-dimensional models were developed to model the multilayered structures and take into consideration the damages. These models were classified in two categories according to the type of the adopted approach. The first category is the mono-layer equivalent approach. The second category, is the layerwise approach.

The mono-layer equivalent approach

It is interesting for engineers that working in plate structures and the researchers working on the creation of plates knowledge, to understand the differences between plate theories and their application.
In the middle of 19th century, many researchers developed the plate theories. These inquiries led to three major categories of plate theory fields.

a) Kirchhoff-Love theory or classical plate theory (CPT), that study thin plate, and neglected the shear effort.

b) Reissner-Mindlin theory, first order shear deformation plate theory (FSDT), that are able to study thick plate, and take into account shear effects.

c) Higher order shear deformation plate theories (HSDT), that study composite plates, and include shear effects.

d) Zig-zag model

a) Kirchhoff-Love theory

Germain, Poisson, Lagrange and Kirchhoff are the first researchers that developed the models of thin plates, by taking inspiration from the theory of beams of Euler-Bernoulli. Based on the properties of Kirchhoff, Love stated a new model of thin plates, named Kirchhoff-Love theory. This theory of thin plates is based on many assumptions. The most important assumption is that the transverse fiber remains straight and perpendicular to the mid-plane after deformation. So, the transverse shear strain is neglected. The Kirchhoff-Love model is appropriate for thin plates. Using this theory gives wrong results when considering thick plates and especially plates made of advanced composites Liu (2011).

b) Reissner-Mindlin theory

To model thick plates, Rayleigh in 1877 and Timoshenko in 1921 show the effect of transverse shear in the study of beams. Reissner (1945) and Uflyand (1948) proposed a first model based on a study of beams, and that takes into account this effect. Then, Mindlin developed his theory in Mindlin (1951).
Many of the hypothesis of the Reissner-Mindlin theory are almost the same as in Kirchhoff-Love theory. The main difference is that the transverse fiber does not remain perpendicular to the mid-plane after deformation. This hypothesis allows to take into consideration the shear effects. Applying Reissner-Mindlin theory for composite laminate shows the difficulty of evaluating shear stress correction factor which is the weakness of this model Shi (2007).

c) Higher-order plate theories

In order to approximate the non-linear distribution of transverse shear stress along the thickness of the plate, many models of higher order shear deformation were developed. These models have been shown to apply to composite plates. Levinson (1980) and Murthy (1981) have developed a variety of theories using polynomial of the third order to establish displacement fields across the plate thickness and without the need for shear correcting factors. They nevertheless used in their theory the equilibrium equations of the classical model, which contradicts the compatibility of the displacements.
In orders, to fix this drawback, Reddy presents in Reddy (1984), the most simple higher-order theory used for composite plate analysis. As Liu (2011) reported, in order to explain the deformation of layered anisotropic plate, Ambartsumian (1958) developed another higher order plate theory based on a transverse shear stress function.
Shi (2007) presented a new HSDT model based on the theories of Murthy and Reddy in 2007. He derived a new set of governing equations associated with proper boundary conditions. In addition, both Reddy and Shi, used 3rd order polynomial to present the displacement component across the plate thickness, leading to parabolic variations of the transverse shear stresses. It should be noticed that it is preferable, whenever it is possible,

not to use, the higher-order plate theories to get the best numerical solution. Indeed, Reddy has written in Reddy (2004), that : " *Higher-order theories can represent the kinematics better, may not require shear correction factors, and can yield more accurate interlaminar stress distributions. However, they involve higher-order stress resultants that are difficult to interpret physically and require considerably more computational effort. Therefore, such theories should be used only when necessary.* "
Figure 3, illustrates the differences between CPT, FSPT, HSDT in terms of in-plane displacement (Wang et al. (2000)).

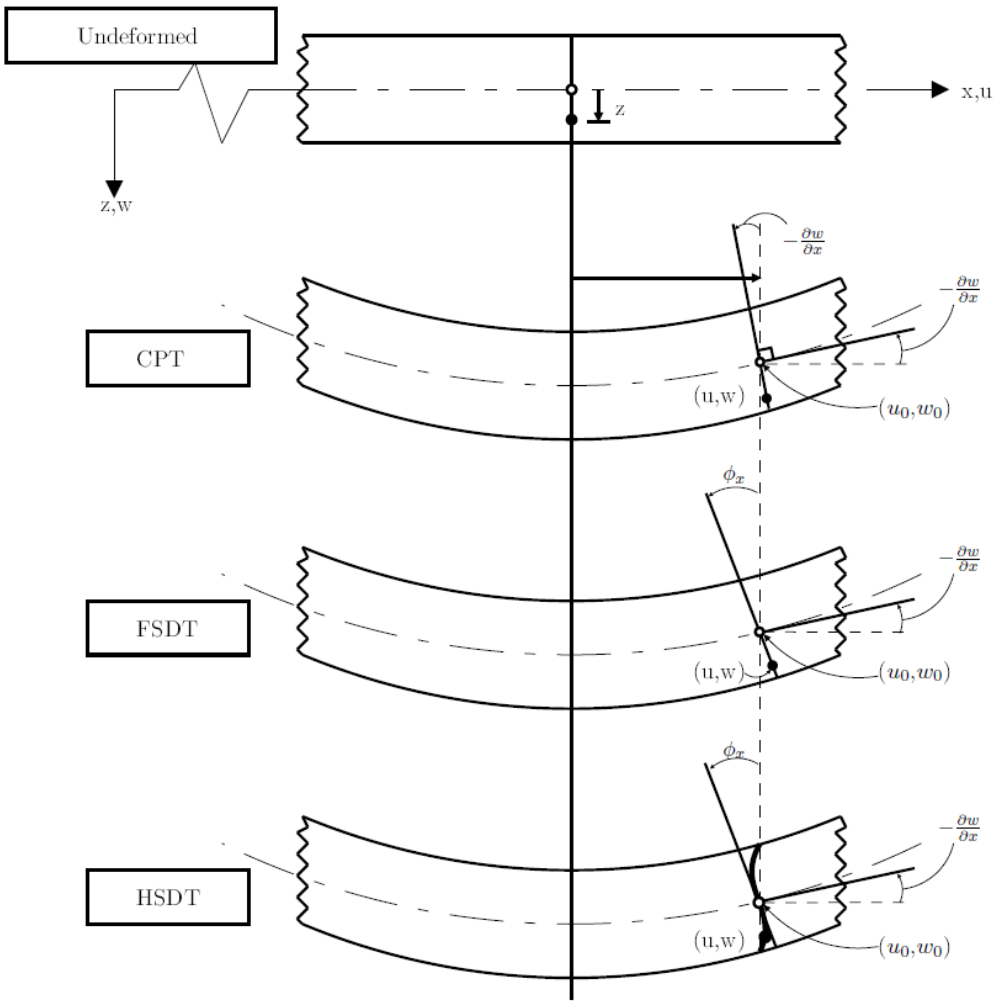


FIGURE 3 – The undeformed and deformed geometries of a plate in various plate theories, where u0 denotes the in-plane displacement(Wang et al. (2000)).

Aydogdu (2006) compares the HSDT models with the three-dimensional analyses available. This study indicated that while the exponential shear deformation theory (Karama

et al. (1998)) predicts well the transverse displacement and stresses, the hyperbolic shear deformation (Soldatos (1992)) and the parabolic shear deformation (Reddy (1984)) theories more suitable for the prediction of natural frequencies and the buckling loads.
Several theories and approaches solve buckling loads, such as Navier method, used to solve laminates with simply supported edges. The Levy (1899) method solutions can be developed for plates with edges having the possibility to combine boundary conditions, and Rayleigh-Ritz method used to calculate approximate solutions for more general boundary conditions.
In general, higher-order models are more accurate then the first order and the classical models. Actually, higher order models presented above are based on different in-plane kinematic functions $f(z)$. Table 1 presents $f(z)$ for some higher-order plate theory.

HSDT	$f(z)$
Shi (2007)	$\frac{5}{4}z\left(1-\frac{4z^2}{3h^2}\right)$
Reddy (1984)	$2\left(1-\frac{4z^2}{3h^2}\right)$
Touratier (1991)	$\frac{h}{\pi}sin\left(\pi\frac{z}{h}\right)$
Karama et al. (1998)	$ze^{-2\left(\frac{z}{h}\right)^2}$

TABLE 1 – Kinematics function $f(z)$ for various higher-order plate theories

d) Zig-zag models

First, we will briefly discuss the zig-zag models. Zig-zag models implement functions which take into account displacement distribution fields at layer interfaces, and because of the difference between layer deformation, the layers are called zig-zag. The interest of zig-zag models is that the displacement fields verify the continuity, without increasing the number of equations. Many researchers tried to improve the zig-zag models(Cho and Parmerter (1993) ; Averill (1994) ; He (1994) ; Incardi (2001) ; Carrera (2004)). However, the analysis of delamination is rarely used in these zig-zag models. Moreover, when the L/h ratio is reduced, the transverse shear prediction is less accurate (Incardi (2001)).

Layerwise approach-Discrete layers models

In order to overcome the drawbacks of the equivalent single layer (ESL), different theories have been proposed.
In discrete layers models, each layer is studied as first-order or higher-order plate and conditions are implemented for displacement and stress in layer interfaces. In our study we are interested in discrete layers models.

Many layerwise models have been proposed by (Barbero and J.N. (1991) ; S.Botello et al. (1999) ; Carrera (1998) ;Moorthya and Reddy (1998) ;Gaudenzi et al. (1995) ;Robbins and Reddy (1993)). In these models each layer is considered as an individual plate, hence the number of equations depends on the number of layers. The layerwise models proved to be a very good substitute for 3D models. The interested readers can refer to (Carrera (2002),2004 ; Zhang and Yang (2009)).
(Crossman and Wang (1982) ; Tahari et al. (1994)) study the delamination and the micro-cracking at the interface between layers. Mainly, these delamination and micro-cracking are caused by the different efforts between the layers. Moreover, (Caron (1993) ; Caron and Ehrlacher (1997)) study the transverse cracking between the layers, caused by the shear efforts at the interfaces.

Benedetti and Milazzo (2017) developed families of equivalent single layerwise and layer-wise models for multilayered plates, taking into account the electrical magnetic fields in the mechanical description.
Transverse anisotropy of multilayered structures make it hard to find closed forms. Hense, it is necessary to use approach solutions in order to study the transverse anisotropy structures. Several authors (Murakami (1985) ; H. (1986) ;Toledano and Murakami (1987a) ; Toledano and Murakami (1987b)) used the Reissner Mixed Variational Theorem (RMVT)(Heillgner-Reissner theory) as a tool to model multilayered structures. Carrera submitted an overall proposal to systematically use RMVT as a way to provide a two-dimensional class of theories for the study of multilayered plates (Carrera (1995) ; Carrera (1997) ; Carrera (1999)). A multilayered RMVT based platform elements that could provide a nearly three-dimensional definition of stresses and strain fields has been developed by (Carrera and Demasi (2002b) ; Carrera and Demasi (2002c) ; Carrera and Demasi (2002a)). Chinosi et al. (2016) demonstrated that the plate elements conditions presented in Chinosi et al. (2013), not only satisfy interlaminar conditions but they also take into account the cross-elasticity effects of anisotropic materials providing good results in terms of transverse shear stresses. Solid models or theories of higher order are needed to capture the localized 3D stress states. However, the drawback of refined plate theories is that it has high computational costs. The Finite Element Method (FEM) plays a dominant role in calculation techniques for layered structural analysis. In Carrera and Zappino (2014), the authors used the Carrera Unified Formulation (CUF) developed in Carrera (2002), Carrera (2003) to develop a beam model with kinematics variables and global/local features. It allows classical to higher order theories to be automatically applied. Then, Carrera et al. (2017) developed a methodology where FEM and Mixed Interpolations of Tensorial Components (MITC) method are adopted (Bathe and Dvorkin (1986) ;Bathe and Brezzi (1987) ;Bathe et al. (1989) ;Bucalem and Dvorkin (1993)). In these works it was proved that global-local models constructed with Node-dependent Kinematics (NDK) can give detailed local effects with fewer computational costs. Zappino et al. (2018) introduced a novel class of 2D FE models for the global-local analysis of multilayered plates, based on a combination of NDK with a p-refinement scheme adopting higher-order Legendre polynomials. Further studies on the added benefit of RMVT for an accurate stress analysis of

multilayered structures in Moleiro et al. (2020).

Moleiro et al. (2019) developed a new layerwise mixed model to statistically analyze multilayer plates with embedded functionally graded materials (FGM) subjected to transverse mechanical loads. This model is developed as an extension of previous work on layerwise mixed models for analysis of multilayered plates (Moleiro et al. (2011) ; Moleiro et al. (2010) ; Moleiro et al. (2012) ;Moleiro et al. (2015)) on layerwise mixed models for analysis of multilayered plates. After that, Gulizzi et al. (2020) presented a novel high-resolution formulation of multilayered composite plates. This formulation is based on the combination of layer-wise plate theories, the discontinuous Galerkin method, and an implicite-defined hierarchical meshing strategy, which allows tuning the resolution of the mechanical fields of interest in suitably selected regions of composite multilayered plates.

A family of discrete layers models, previously called Multi-particle model of multilayered materials (M4), are developed in Navier. Later on in accordance with existing terminology in literature they will renamed layerwise stress approach with first-order membrane stress approximation per layer in the thickness direction (LS1) model.
By taking inspiration from (Pagano (1978)) many layerwise stress models (Chabot (1997) ; Carreria et al. (2002) ; Hadj-Ahmed et al. (2001) ; Diaz Diaz (2001) Dallot and Sab (2008) ;Diaz et al. (2002) ;Lerpiniere et al. (2014) ;Nguyen and Caron (2006) ;Saeedi et al. (2012a), 2012b, 2013a, 2013b ; Baroud et al. (2016)) were built from the variational formulation of Hellinger-Reissner and a polynomial approximation of the stresses fields by layer. In these models, the membrane stresses are first degree polynomials, the shear stresses fields are quadratic by layer, and the normal stresses are cubic polynomials. A family of a stress fields approaches has been proposed in order to develop these models. The generalized strain dual of the generalized stress is deduced by introducing these stress fields in the Hellinger-Reissner functional. The equilibrium equations and the boundary conditions are obtained by considering the variations of this functional with respect to the variation of generalized displacements fields. In addition, the constitutive laws are obtained by considering the variations of the Hellinger-Reissner functional with respect to generalized stresses.
These models were named Multi-particle models of multilayered plates (M4). Because there are n particles at each geometric point of the 2D plate, where n is the number of layers and each particle has a different kinematics.
The (M4) modelling approach avoids singularities and reduces the real model by one dimension. These features accelerate the resolution of the system with respect to other models.
Chabot (1997), studied the multiparticular models of multilayered plates in her PhD thesis. She developed six models to analyse stress at the interfaces in the multilayered plates. The interested reader can refer to (Chabot and Ehrlacher (1998a)). In this study, Chabot started from the model M4-7n which is similar to the Pagano model (Pagano (1978)). By simplification and elimination of some generalized stresses, she reached a simplest

multi-particular model M4-(2n+1)M as shown in figure 4.

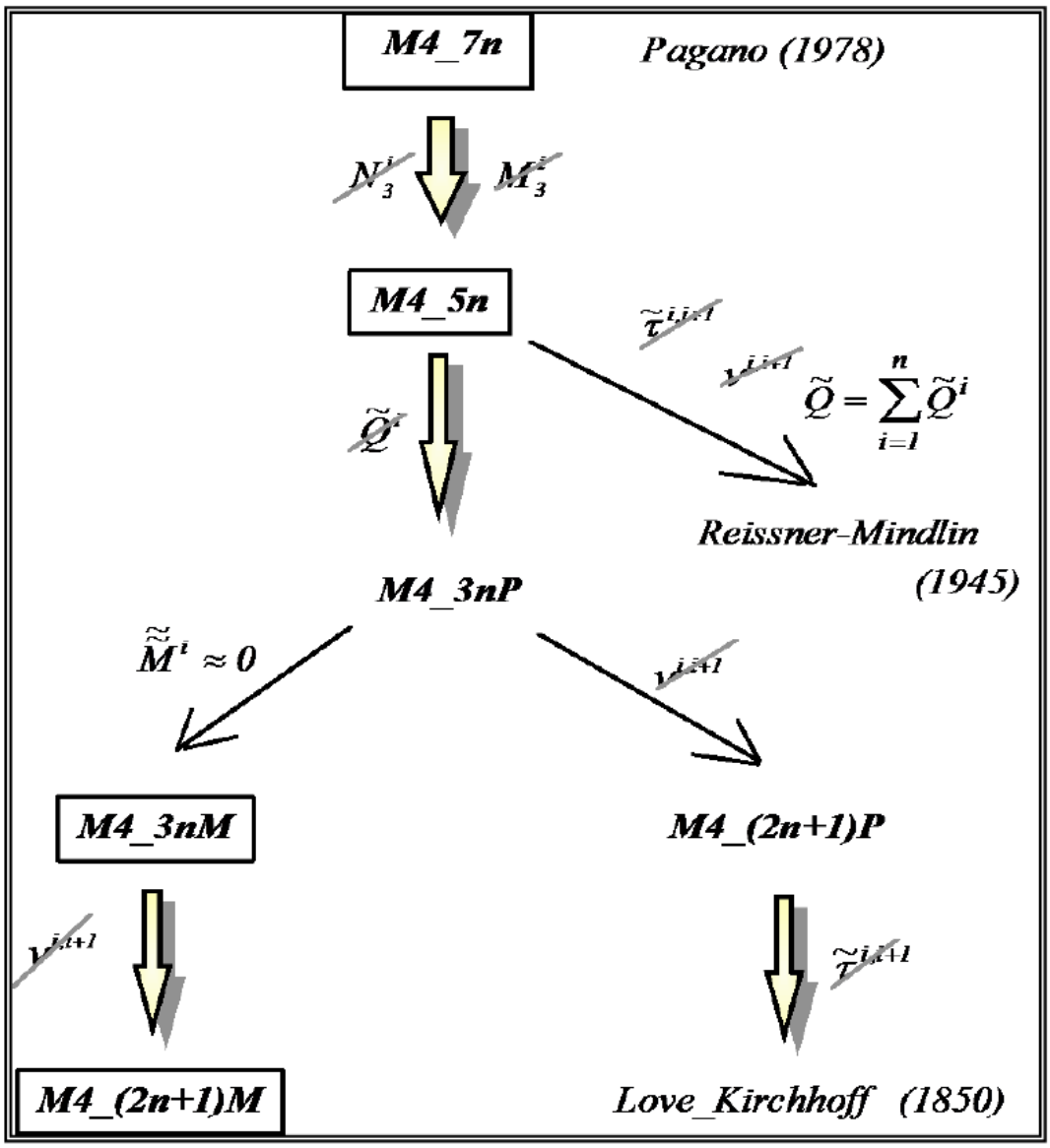


FIGURE 4 – Derivation of multiparticulate models (M4) (Chabot (1997)) (Chabot and Ehrlacher (1998a))

M4-(2n+1)M is a generalization of the classic-shear-lag analysis (Garett and Bailley (1977)). However, it does not give a good prediction of the 3D stresses, and leads to a linear pulling force at the edge of the interface, which appears to be directly related to the rate of energy relaxation in a possible phenomenon of delamination.
On the other hand, the model labelled M4-5n (one of the M4 models) seems able to appropriately reproduce the stresses at the interfaces. The solutions of M4-5n equations were

validated by comparision with the 3D finite element method by (Hun et al. (2011) ; Carreira (1998) ; Carreria et al. (2002)). The concept of M4-5n was generalized in Diaz Diaz (2001),Caron et al. (2006), Diaz et al. (2007). In addition, a quasi-analytical solution of the stress fields and energy rate has been developed Chabot et al. (2013), and validated by a comparison with the results of Dundurs (1969) and Hun (2012). A semi-analytical elastic solution was presented by Nasser and Chabot (2018) based on Nasser's work presented in Nasser and Chabot (2015). It gives a parametric study of 2D cracked pavements resting on a soil. The location of the loads on the cracks is confirmed by Nasser and Chabot (2018) for the study of 2D paves. The authors improved soil simulation so that the mechanical fields between the layers were better approximated. Nasser et al. (2016) developed a M4-5n model that can be used to simulate a plate geometry with cracks and delamination between layers. Moreover a numerical tool for computing the mechanical response of pavements containing vertical cracks and interlayer debonding is presented by Nasser et al. (2018a).

Furthermore, a simplified mutli-particule model of multilayered plates named M4-5nB (Boussinesq), was proposed in (Chabot et al. (2005) ; Chabot et al. (2003), 2004 ; Chabot et al. (2004a), 2004b, 2007). In this model the authors used a software for the 3D evolution of cracks in pavements. The advantage of such model is that it reduces the studied model by one dimension, and gives very accurate approximations of the stress fields near the interfaces, cracks and edges when comparing with 3D finite elements calculation and is less time consuming. (Chabot et al. (2007)) used the M4-5nB model to study the propagation of vertical crack in pavements.
In accordance with (Deep (2017) and Chabot and Deep (2018)), Chabot and Deep (2019) presented new developments in the M4-5nW model that allow considering a material discontinuity in one or several materials layers.

Even though the LS1 model and its refined version are very effective, they can still be improved. These models cannot exactly fulfill the 3D stress-free boundary conditions. Moreover, since these models are based on mixed Hellinger-Reissner variational theory, there is no theoretical guarantee that the refined LS1 model will converge to the 3D model if the number of mathematical layers increases. Baroud et al. (2016) implemented a layerwise model called Statically Compatible LS1 (SCLS1) or M4-(6n+1) to boost the LS1 model by eliminating its disadvantages. SCLS1 is a multilayered model for linear elastic plates. It derives from the exact 3D model by taking into account the statically compatible stresses in the layer direction of thickness with first order membrane stress approximations per layer. In addition to interlaminar shear and normal stress on the interfaces between layers and variations in interlaminar shear stress, the generalized stresses of the SCLS1 model takes into account the the layer-by-layer stresses of Reissner-Mindlin plate.
The next chapter is dedicated to the detailed description on the SCLS1 model.

Chapitre 1

The Statically Compatible Layerwise model SCLS1

In this chapter the benefits of using the SCLS1 model will be presented, based on Baroud's study Baroud (2016). We will introduced the equations of the existing SCLS1 model for elastic multilayered plates as seen in Baroud et al. (2016). We will give the definitions of the generalized strains and stresses of this model, as well as the equilibrium equations and the constitutive equations. The generalized stresses of the SCLS1 model are the generalized stresses of the Reissner-Mindlin plate per layer in addition to the inter-laminar shear and normal stresses at the interfaces between layers and the divergences of this inter-laminar shear stresses. Moreover, we have 6n-1 equilibrium equations in terms of the generalized stresses where the number of layers n was derived from the 3D exact equilibrium equations. Finally, the relation between the generalized stresses and generalized strains was given by the generalized constitutive equations deduced by using the stress energy formulation.

1.1 Notations and model description

We consider a linear elastic multilayered plate composed of n orthotropic elastic layers and occupying the $3D$ domain $\Omega = \omega \times]h_1^-, h_n^+[$ where $\omega \subset \mathbb{R}^2$ is the middle surface of the plate and h its thickness. The plate is subjected to forces in its upper face ω^+ and lower face ω^- with the distributed surface forces $T^+ = (T_k^+)$ and $T^- = (T_k^-)$. The boundary of the domain, denoted by $\partial\Omega$, is decomposed into two parts : a free part $\partial\Omega_N = \partial\omega_N \times]h_1^-, h_n^+[$ where $T = (T_k) = (\sigma_{kl} n_l)$ is set to zero, and a restrained part $\partial\Omega_D = \partial\omega_D \times]h_1^-, h_n^+[$ where the displacement $u = (u_k)$ is set to zero. The subset $\partial\omega_N$ and $\partial\omega_D$ are the partition of $\partial\omega$, and $n = (n_k)$ is the outer normal of $\partial\omega_N$.

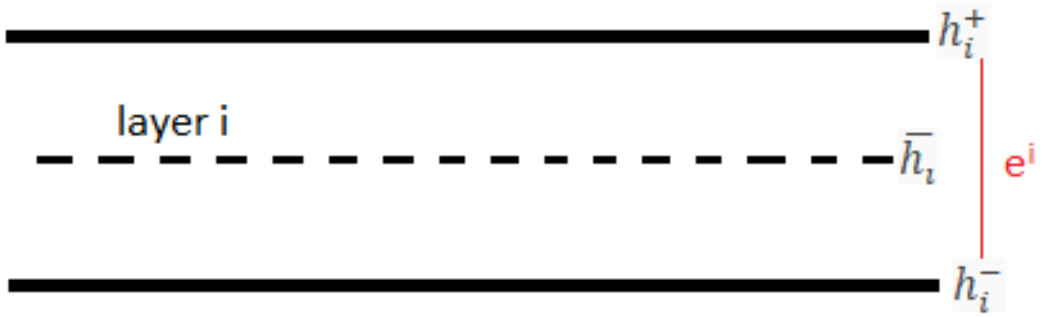


FIGURE 1.1 – Notations in a layer

In the following, x and y are the in-plane coordinates and z is the out-of-plane coordinate. The following notations are introduced :

- The subscripts i and $j, j+1$ indicate layer i and the interface between layers j and $j+1$ with $1 \leq i \leq n$ and $1 \leq j \leq n-1$, respectively. By extension, the

superscripts $0, 1$ and $n, n+1$ refers to the lower face $\omega^- = \omega \times h_1^-$ and the upper face $\omega^+ = \omega \times h_n^+$ respectively.

- In each layer i, h_i^-, h_i^+ and $\overline{h}_i$ are, respectively, the bottom, the top and the mid-plane z coordinate of the layer, and $e^i = h_i^+ - h_i^-$ is the thickness. Thus we have $h_i^+ = h_{i+1}^-$ for all $a \leq i \leq n-1$, we set $h_0^+ = h_1^-$ and $h_{n+1}^- = h_n^+$.
- Greek subscripts $\alpha, \beta, \gamma, \cdots \in \{1, 2\}$ indicate the in-plane components.
- Latin subscripts $k, l, m, n, \cdots \in \{1, 2, 3\}$ indicate the $3D$ components.
- ${}^t[X]$ is the transpose of $[X]$.
- $(\boldsymbol{S}^i = (S^i_{klmn}))$ is the fourth-order 3D compliance tensor of layer i with the minor and major symmetries : $S^i_{klmn} = S^i_{lkmn} = S^i_{klnm} = S^i_{mnkl}$ and it is positive definite. Its inverse is the 3D elasticity stiffness tensor and is denoted by (C^i_{klmn}) for layer i. The tensor (C^i_{klmn}) possesses the same symmetries as (S^i_{klmn}) and it is also positive definite.
- $\boldsymbol{S}^i$ is monoclinic in direction z : $S^i_{\alpha\beta\gamma 3} = S^i_{\alpha 333} = 0$
- $\sigma_{\alpha\beta}(x, y, z)$ are the in-plane stress components, $\sigma_{\alpha 3}(x, y, z)$ are the transverse shear stresses and $\sigma_{33}(x, y, z)$ is the normal stress.
- $\epsilon_{\alpha\beta}(x, y, z)$ are the in-plane strain components, $\epsilon_{\alpha 3}(x, y, z)$ are the transverse strain stresses and $\epsilon_{33}(x, y, z)$ is the normal strain.
- $u_\alpha(x, y, z)$ are the in-plane 3D displacement components, $u_3(x, y, z)$ is the normal 3D displacement component.

1.2 Governing equations of the 3D model

The 3D elastic problem is to find a statically compatible stress field $\boldsymbol{\sigma} = (\sigma_{kl})$, and a kinematically strain field $\boldsymbol{\epsilon} = (\epsilon_{kl})$ which comply with the constitutive equation :

$$\epsilon_{kl}(x, y, z) = S_{klmn}(z) : \sigma_{mn}(x, y, z) \text{ on } \Omega, \tag{1.1}$$

where the stress field $\boldsymbol{\sigma}$ is statically compatible if it complies with the equilibrium equations :

$$\sigma_{kl,l} = 0 \text{ on } \Omega, \tag{1.2}$$

and the stress conditions on the upper and the lower faces :

$$\sigma_{k3} = -T_k^- \text{ on } \omega^-, \ \sigma_{k3} = T_k^+ \text{ on } \omega^+, \tag{1.3}$$

and on the lateral boundary :

$$\sigma_{kl} n_l = 0 \text{ on } \partial\Omega_N. \tag{1.4}$$

A strain field $\boldsymbol{\epsilon}$ is kinematically compatible if there exists a displacement field $\boldsymbol{u} = (u_k)$ complying with the displacement conditions on the lateral boundary :

$$u_k = 0 \text{ on } \partial\Omega_D, \tag{1.5}$$

and such that,

$$\epsilon_{kl} = \frac{1}{2}\left(u_{k,l} + u_{l,k}\right) \text{ on } \Omega. \tag{1.6}$$

1.3 The static of SCLS1 model

In this section, we construct an approximation of $3D$ stress fields, as z polynomial forms per layer. By taking into consideration the $3D$ equilibrium equations, if the polynomial degree of the membrane components $\sigma_{\alpha\beta}$ of the $3D$ stress fields is d, then the degree of the transverse shear stresses $\sigma_{\alpha 3}$ is at most d+1 and that of the normal stress σ_{33} is at most d+2.
The SCLS1 model considers the following form of the $3D$ stresses in layer i, for $1 \leq i \leq n$, such that $\sigma_{\alpha\beta}$ are layerwise linear functions of z. By integrating the first two equations of the $3D$ equilibrium equations we conclude that the transverse shear stress $\sigma_{\alpha 3}$ are layerwise quadratic functions of z. In addition, by integrating the last equation of the $3D$ equilibrium equations we can find that the normal stress σ_{33} is a layerwise third-order polynomial function of z. The stresses $\sigma_{\alpha 3}$ and σ_{33} take into account the continuity at the interfaces between the layers. The coefficients of these polynomials are the generalized stresses. Moreover, this from of the $3D$ stress field is the one and only one hypothesis of the SCLS1 model :

$$\sigma^{3D}_{\alpha\beta}(x,y,z) = N^i_{\alpha\beta}(x,y)\frac{P^i_0(z)}{e^i} + \frac{12}{{e^i}^2}M^i_{\alpha\beta}(x,y)P^i_1(z), \tag{1.7}$$

$$\begin{aligned} \sigma^{3D}_{\alpha 3}(x,y,z) &= Q^i_\alpha \frac{P^i_0(z)}{e^i} + \left(\boldsymbol{\tau}^{i,i+1}_\alpha(x,y) - \boldsymbol{\tau}^{i-1,i}_\alpha(x,y)\right) P^i_1(z) \\ &+ \left(Q^i_\alpha(x,y) - \frac{e^i}{2}\left(\boldsymbol{\tau}^{i,i+1}_\alpha(x,y) + \boldsymbol{\tau}^{i-1,i}_\alpha(x,y)\right)\right)\frac{P^i_2(z)}{e^i}, \end{aligned} \tag{1.8}$$

(1.9)

$$\begin{aligned}
\sigma_{33}^{3D}(x,y,z) &= \left(\frac{1}{2}\left(\nu^{i,i+1}(x,y)+\nu^{i-1,i}(x,y)\right)+\frac{e^i}{12}\left(\pi^{i,i+1}(x,y)-\pi^{i-1,i}(x,y)\right)\right)P_0^i(z)\\
&+ \left(\frac{6}{5}\left(\nu^{i,i+1}(x,y)-\nu^{i-1,i}(x,y)\right)+\frac{e^i}{10}\left(\pi^{i,i+1}(x,y)+\pi^{i-1,i}(x,y)\right)\right)P_1^i(z)\\
&+ \left(\frac{e^i}{12}\left(\pi^{i,i+1}(x,y)-\pi^{i-1,i}(x,y)\right)\right)P_2^i(z)\\
&+ \left(\frac{e^i}{2}\left(\pi^{i,i+1}(x,y)+\pi^{i-1,i}(x,y)\right)+\left(\nu^{i,i+1}(x,y)-\nu^{i-1,i}(x,y)\right)\right)P_3^i(z),
\end{aligned}$$

where P_k^i, $k=0,1,2,3$, are the orthogonal Legendre-like polynomial basis defined on layer i by : for $h_i^- \le z \le h_i^+$,

(1.10)

$$\begin{cases}
P_0^i(z) &= 1\\
P_1^i(z) &= \dfrac{z-\overline{h}_i}{e^i}\\
P_2^i(z) &= -6\left(\dfrac{z-\overline{h}_i}{e^i}\right)^2+\dfrac{1}{2}\\
P_3^i(z) &= -2\left(\dfrac{z-\overline{h}_i}{e^i}\right)^3+\dfrac{3}{10}\left(\dfrac{z-\overline{h}_i}{e^i}\right)
\end{cases}$$

In order to facilitate the calculation of the elastic energy, these polynomials are chosen orthogonal to each other.
We can express these polynomials using the classical Legendre polynomials :

$$P_0^L(x)=1 \quad P_1^L(x)=x \quad P_2^L(x)=\frac{3x^2-1}{2} \quad P_3^L(x)=\frac{5x^3-3x}{2},$$

The orthogonal Legendre-like polynomial basis are then given by these relations :

$$\begin{cases}
P_0^i(x) &= P_0^L\left(2\left(\dfrac{z-\overline{h}_i}{e^i}\right)\right)\\
P_1^i(x) &= \dfrac{1}{2}P_1^L\left(2\left(\dfrac{z-\overline{h}_i}{e^i}\right)\right)\\
P_2^i(x) &= -P_2^L\left(2\left(\dfrac{z-\overline{h}_i}{e^i}\right)\right)\\
P_3^i(x) &= -\dfrac{1}{10}P_3^L\left(2\left(\dfrac{z-\overline{h}_i}{e^i}\right)\right)
\end{cases}$$

and where,

- $N^i = (N_{\alpha\beta})$ is the in-plane stress resultants tensor, related to the 3D local stress (σ_{ij}) in each layer i by :

$$N^i = (N^i_{\alpha\beta}) = \langle \sigma^{3D}_{\alpha\beta} \rangle,$$

  where the integration through the thickness is noted $\langle \cdot \rangle : \int_{h_i^-}^{h_i^+} f(z)dz = \langle f \rangle$.
- $M^i = (M_{\alpha\beta})$ is the moment resultants tensor expressed in terms of the 3D stress field σ^{3D} in each layer i as follows :

$$M^i = (M^i_{\alpha\beta}) = \langle (z - \overline{h}_i)\sigma^{3D}_{\alpha\beta} \rangle.$$

- $Q^i = (Q^i_\alpha)$ is the out-of-plane shear stress resultant vector, defined from the 3D stress field σ^{3D} in each layer i as follows :

$$Q^i = (Q^i_\alpha) = \langle \sigma^{3D}_{\alpha 3} \rangle.$$

- $\tau^{j,j+1}_\alpha$ is the interlaminar shear stress at the interface between layer j, and layer $j+1$ for $0 \leq j \leq n-1$ given by :

$$\tau^{j,j+1}_\alpha(x,y) = \sigma^{3D}_{\alpha 3}(x,y,h^+_j) = \sigma^{3D}_{\alpha 3}(x,y,h^-_{j+1}).$$

- $\nu^{j,j+1}$ is the normal stress at the interface between j and $j+1$, for $0 \leq j \leq n-1$, given by :

$$\nu^{j,j+1}(x,y) = \sigma^{3D}_{33}(x,y,h^+_j) = \sigma^{3D}_{33}(x,y,h^-_{j+1}).$$

- $\pi^{j,j+1}$ is an additional parameter which allows to consider σ_{33} as a polynomial of third degree per layer. By writing the generalized equilibrium equations we can find that $\pi^{j,j+1}$ is the divergence of the interlaminar shear stress vector $\boldsymbol{\tau}^{j,j+1} = (\boldsymbol{\tau}^{j,j+1}_\alpha)$ defined on the interface between layers j and $j+1$ for $0 \leq j \leq n-1$.

1.4 The equilibrium equations

The $3D$ stress field $\boldsymbol{\sigma}^{3D}$ shall satisfy 3D equilibrium equations (1.2), if and only if, the following equations hold true for all (x,y) in ω and for all $i = 1,\ldots,n$ and $j = 0,\ldots,n$:

$$
\begin{aligned}
&N^i_{\alpha\beta,\beta} + \tau^{i,i+1}_\alpha - \tau^{i-1,i}_\alpha = 0.\\
&M^i_{\alpha\beta,\beta} - Q^i_\alpha + \frac{e^i}{2}(\tau^{i,i+1}_\alpha + \tau^{i-1,i}_\alpha) = 0.\\
&Q^i_{\beta,\beta} + \nu^{i,i+1} - \nu^{i-1,i} = 0.\\
&\tau^{j,j+1}_{\beta,\beta} - \pi^{j,j+1} = 0.
\end{aligned}
\tag{1.11}
$$

The last equation gives the interpretation of $\pi^{j,j+1}$ which is equal to the divergence of the interlaminar shear stress vector $\tau^{j,j+1} = (\tau^{j,j+1}_\alpha)$. Now the stress boundary conditions have also to be enforced in addition to the equations (1.11). The lateral boundary conditions $\sigma^{3D}_{ij} n_j = 0$ on $\partial\Omega_N$ are equivalent to the following equations for $i = 1, \ldots, n$ and $j = 0, \ldots, n$:

$$
N^i_{\alpha\beta} n_\beta = 0,\ M^i_{\alpha\beta} n_\beta = 0,\ Q^i_\alpha n_\alpha = 0,\ \tau^{j,j+1}_\alpha n_\alpha = 0,\ \text{on } \partial\omega_N.
\tag{1.12}
$$

The boundary conditions (1.3) on the upper an the lower faces write, respectively,

$$
\begin{cases}
\tau^{0,1}_1(x,y) &= -T^-_1(x,y),\\
\tau^{0,1}_2(x,y) &= -T^-_2(x,y),\\
\nu^{0,1}(x,y) &= -T^-_3(x,y).
\end{cases}
\quad \text{and} \quad
\begin{cases}
\tau^{n,n+1}_1(x,y) &= T^+_1(x,y),\\
\tau^{n,n+1}_2(x,y) &= T^+_2(x,y),\\
\nu^{n,n+1}(x,y) &= T^+_3(x,y).
\end{cases}
\tag{1.13}
$$

It should be noted that the boundary conditions (1.12) and (1.13) cannot be simultaneously verified unless $T^\pm_\alpha n_\alpha = 0$ on $\partial\omega_N$, which will be assumed in the sequel. Moreover, from the last equations of (1.11) for $j = 0$ and $j = n$, we see that :

$$
\pi^{0,1} = -T^-_{\alpha,\alpha} \text{ and } \pi^{n,n+1} = T^+_{\alpha,\alpha}.
\tag{1.14}
$$

Finally, the stress field $\boldsymbol{\sigma}^{3D}$ is statically compatible when it complies with the generalized equilibrium equations on ω : (1.11) for $i = 1, \ldots, n$ and $j = 1, \ldots, n-1$, (1.13) and (1.14), and with the generalized stress free boundary conditions on $\partial\omega_N$: (1.12) for $i = 1, \ldots, n$ and $j = 1, \ldots, n-1$.

1.5 Generalized displacements and strains

The SCLS1 generalized displacements are ($U^i_\alpha, U^i_3, \Phi^i_\alpha$ and $V^{j,j+1}$). U^i_α is the two in-plane displacements, U^i_3 is the vertical displacement, Φ^i_α is the two bending rotations in each layer i and $V^{j,j+1}$ is a kinematical variable, having the dimension of an area, which is dual of the static variable $\pi^{j,j+1}$ defined on the interface $j, j+1$.

$$U^i_\alpha(x,y) = \int_{h_i^-}^{h_i^+} \frac{P^i_0(z)}{e^i} u_\alpha(x,y,z)dz, \tag{1.15}$$

$$\Phi^i_\alpha(x,y) = \int_{h_i^-}^{h_i^+} \frac{12}{{e^i}^2} P^i_1(z) u_\alpha(x,y,z)dz, \tag{1.16}$$

$$U^i_3(x,y) = \int_{h_i^-}^{h_i^+} \left(\frac{P^i_0(z)}{e^i} + \frac{P^i_2(z)}{e^i} \right) u_3(x,y,z)dz, \tag{1.17}$$

$$W^i_\pm(x,y) = \int_{h_i^-}^{h_i^+} \left(P^i_1(z) \pm \frac{P^i_2(z)}{2} \right) u_3(x,y,z)dz, \tag{1.18}$$

and,

$$V^{j,j+1}(x,y) = W^j_-(x,y) - W^{j+1}_+(x,y). \tag{1.19}$$

In addition, the generalized displacement, verifies the following generalized boundary conditions for $1 \leq i \leq n$ and $0 \leq j \leq n$:

$$U^i_\alpha = 0, \quad U^i_3 = 0, \quad \Phi^i_\alpha = 0, \quad V^{j,j+1} = 0, \qquad \text{on } \partial\omega_D. \tag{1.20}$$

The generalized strains dual of the generalized stresses $N^i_{\alpha\beta}, M^i_{\alpha\beta}, \tau^{j,j+1}_\alpha, \nu^{j,j+1}, \pi^{j,j+1}$ for $i = 1, \cdots, n$ and $j = 1, \cdots, n-1$ are respectively expressed in terms of the generalized displacements as :

$$
\begin{aligned}
\epsilon^i_{\alpha\beta} &= \frac{1}{2}\left(U^i_{\alpha,\beta} + U^i_{\beta,\alpha}\right), \\
\chi^i_{\alpha\beta} &= \frac{1}{2}\left(\Phi^i_{\alpha,\beta} + \Phi^i_{\beta,\alpha}\right), \\
\gamma^i_\alpha &= \Phi^i_\alpha + U^i_{3,\alpha}, \\
D^{j,j+1}_\alpha &= U^{j+1}_\alpha - U^j_\alpha - \frac{e^j}{2}\Phi^j_\alpha - \frac{e^{j+1}}{2}\Phi^{j+1}_\alpha, \\
D^{j,j+1}_\nu &= U^{j+1}_3 - U^j_3, \\
\lambda^{j,j+1} &= V^{j,j+1}.
\end{aligned}
\tag{1.21}
$$

1.6 The constitutive equations of SCLS1 model

The $3D$ elastic energy of the $3D$ stress field $\boldsymbol{\sigma}^{3D}$ was given in the following from :

$$
\begin{aligned}
W^*_{3D} &= \int_\Omega \frac{1}{2} S_{klmn}(z)\sigma^{3D}_{mn}\sigma^{3D}_{kl}dxdydz \\
&= \int_\omega w^*_{3D}dxdy
\end{aligned}
$$

where w^*_{3D} is the generalized stress energy density per unit area of the plate defined by :

$$
w^*_{3D} = \int_{h_1^-}^{h_n^+} \frac{1}{2} S_{klmn}\sigma^{3D}_{mn}\sigma^{3D}_{kl}dz.
$$

In Multi-particle models for multilayered plates as SCLS1 model, to construct the compliance matrix, it is enough to derive the generalized energy density with respect to each generalized stresses. We will then we obtain the generalized constitutive equations.

The (3n + 3(n-1)) constitutive equations of the SCLS1 model where 3 are over the layer i and 3 on the interfaces, expressed using the stress energy associated to $\boldsymbol{\sigma}^{3D}$ are given by : for $1 \leq i \leq n$ and for $1 \leq j \leq n-1$

- Membrane constitutive equation of layer i :

$$
\epsilon^i_{\alpha\beta} = \frac{1}{e^i} S^i_{\alpha\beta\gamma\delta} N^i_{\gamma\delta} + S^i_{\alpha\beta 33}\left(\frac{1}{2}\left(\nu^{i,i+1} + \nu^{i-1,i}\right) + \frac{e^i}{12}\left(\pi^{i,i+1} - \pi^{i-1,i}\right)\right).
\tag{1.22}
$$

- Bending constitutive equations of layer i :

$$\chi^i_{\alpha\beta} = \frac{12}{{e^i}^3} S^i_{\alpha\beta\gamma\delta} M^i_{\gamma\delta} + \frac{1}{e^i} S^i_{\alpha\beta 33} \left(\frac{6}{5} \left(\nu^{i,i+1} - \nu^{i-1,i} \right) + \frac{e^i}{10} \left(\pi^{i,i+1} + \pi^{i-1,i} \right) \right). \tag{1.23}$$

- Transverse shear constitutive equation of layer i :

$$\gamma^i_\alpha = \frac{24}{5e^i} S^i_{\alpha 3\beta 3} Q^i_\beta - \frac{2}{5} S^i_{\alpha 3\beta 3} \left(\tau^{i,i+1}_\beta + \tau^{i-1,i}_\beta \right). \tag{1.24}$$

- Shear constitutive equation of interface $j, j+1$:

$$\begin{aligned} D^{j,j+1}_\alpha &= -\frac{2}{5} S^j_{\alpha 3\beta 3} Q^j_\beta - \frac{2}{5} S^{j+1}_{\alpha 3\beta 3} Q^{j+1}_\beta - \frac{2}{15} e^j S^j_{\alpha 3\beta 3} \tau^{j-1,j}_\beta \\ &\quad + \frac{8}{15} \tau^{j,j+1}_\beta \left(e^j S^j_{\alpha 3\beta 3} + e^{j+1} S^{j+1}_{\alpha 3\beta 3} \right) - \frac{2}{15} e^{j+1} S^{j+1}_{\alpha 3\beta 3} \tau^{j+1,j+2}_\beta. \end{aligned} \tag{1.25}$$

- Normal constitutive equation of interface $j, j+1$:

$$\begin{aligned} D^{j,j+1}_\nu &= \frac{9}{70} e^j S^j_{3333} \nu^{j-1,j} + \frac{13}{35} \left(e^j S^j_{3333} + e^{j+1} S^{j+1}_{3333} \right) \nu^{j,j+1} \\ &\quad + \frac{9}{70} e^{j+1} S^{j+1}_{3333} \nu^{j+1,j+2} - \frac{13}{420} \left(e^j \right)^2 S^j_{3333} \pi^{j-1,j} \\ &\quad + \frac{11}{210} \left(\left(e^j \right)^2 S^j_{3333} - \left(e^{j+1} \right)^2 S^{j+1}_{3333} \right) \pi^{j,j+1} \\ &\quad + \frac{13}{420} (e^{j+1})^2 S^{j+1}_{3333} \pi^{j+1,j+2} + \frac{1}{2} S^j_{\alpha\beta 33} N^j_{\alpha\beta}. \\ &\quad + \frac{1}{2} S^{j+1}_{\alpha\beta 33} N^{j+1}_{\alpha\beta} + \frac{6}{5e^j} S^j_{\alpha\beta 33} M^j_{\alpha\beta} - \frac{6}{5e^{j+1}} S^{j+1}_{\alpha\beta 33} M^{j+1}_{\alpha\beta}. \end{aligned} \tag{1.26}$$

- Constitutive equation for the π generalized stress at interface $j, j+1$:

$$\begin{aligned} \lambda^{j,j+1} &= \frac{1}{105} \left(S^j_{3333} \left(e^j \right)^3 + \left(e^{j+1} \right)^3 S^{j+1}_{3333} \right) \pi^{j,j+1} - \frac{1}{140} S^j_{3333} \left(e^j \right)^3 \pi^{j-1,j} \\ &\quad - \frac{1}{140} \left(e^{j+1} \right)^3 S^{j+1}_{3333} \pi^{j+1,j+2} + \frac{11}{210} \nu^{j,j+1} \left(\left(e^j \right)^2 S^j_{3333} - \left(e^{j+1} \right)^2 S^{j+1}_{3333} \right) \\ &\quad + \frac{13}{420} \left(e^j \right)^2 S^j_{3333} \nu^{j-1,j} - \frac{13}{420} \left(e^{j+1} \right)^2 S^{j+1}_{3333} \nu^{j+1,j+2} \\ &\quad + \frac{e^j}{12} S^j_{\alpha\beta 33} N^j_{\alpha\beta} - \frac{e^{j+1}}{12} S^{j+1}_{\alpha\beta 33} N^{j+1}_{\alpha\beta} + \frac{1}{10} S^j_{\alpha\beta 33} M^j_{\alpha\beta} + \frac{1}{10} S^{j+1}_{\alpha\beta 33} N^{j+1}_{\alpha\beta}. \end{aligned} \tag{1.27}$$

1.7 Benefits of SCLS1 model

The SCLS1 model is derived from the 3D exact model by considering Statically Compatible Layerwise Stresses with first-order membrane stress approximations per layer in the thickness direction. This model was developed to fix the drawbacks of LS1 model.As in the LS1 model, the plate is considered as an overlay of Reissner plates connected by interfacial stresses. The SCLS1 model has new generalized stresses which are the divergence of the interlamminar transverse shears at the interface between layers.

Baroud et al. (2016), showed many examples in which he studied the stress distribution. In this paragraph, we will present the example of the rectangular laminated with a circular hole placed in the center of the plate. Baroud studied a $(90°, 0°, 90°)$ laminated with a circular hole and subjected to a tensile load, where the thickness of all the layers is equal $(e_1 = e_2 = e_3 = e = 1mm)$. We show in figure 1.2 the laminate of three-ply of thickness $3e$, a length of $2l = 160e$, a width of $2b = 80e$ and radius of the central hole is equal to $R = 10e$. In addition, this laminated is subjected to a uniform displacement in the x direction.

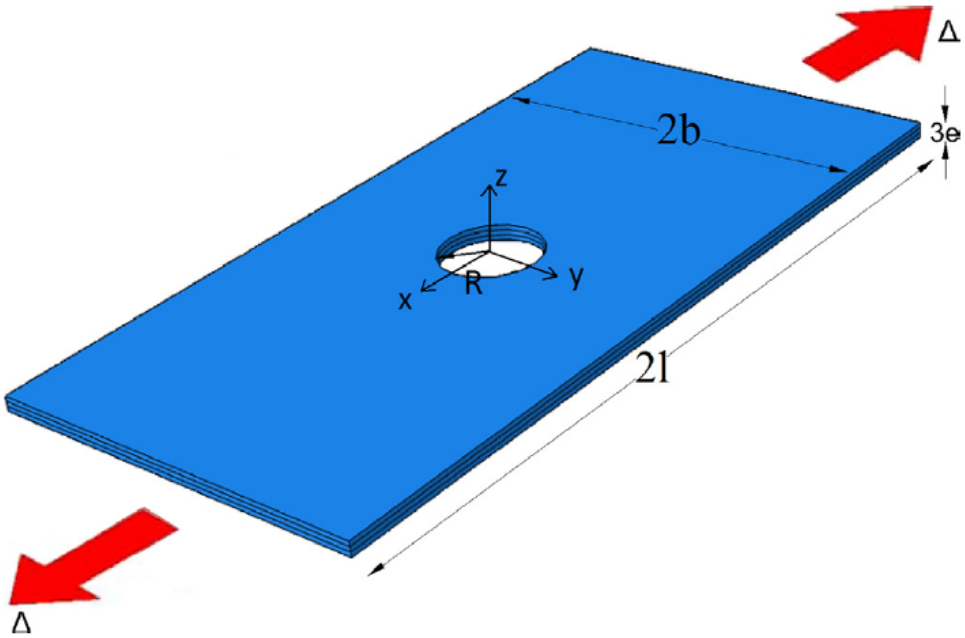


FIGURE 1.2 – The laminate with circular hole and loading presented in (Baroud et al. (2016))

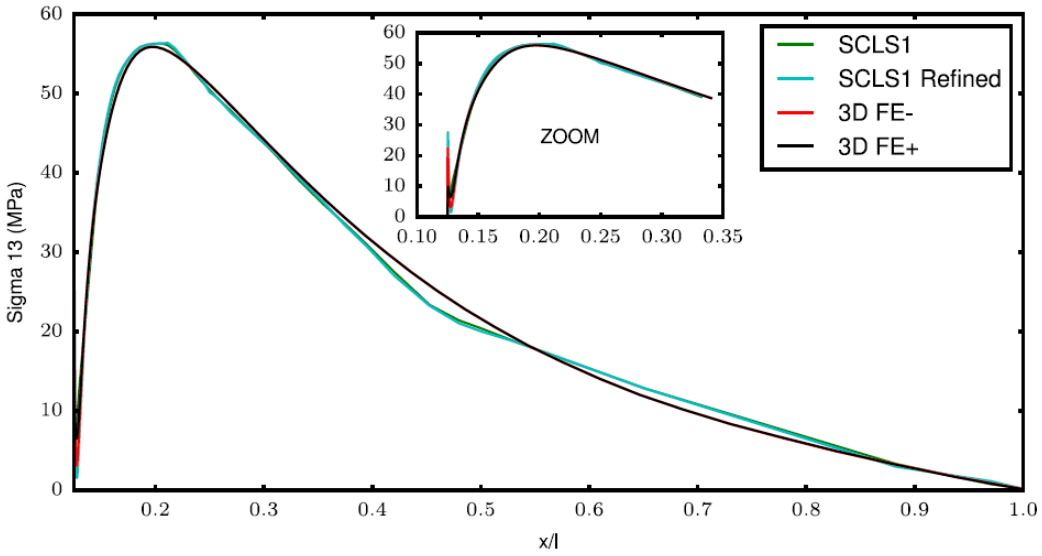


FIGURE 1.3 – The shear stress σ_{13} at the interface between layers 1 and 2 presented in (Baroud et al. (2016))

The figure 1.3 present the shear interlaminar stress σ_{13} for $y = 0$ at the interface between layer 1 and layer 2 (90°/0°) for the SCLS1, the Refined version of SCLS1 and the 3D finite element. The FE^- (resp. $FE+$) presents the shear stress σ_{13} at the interface 90°/0° in the 90° layer (resp. in the 0° layer).
It is noteworthy that the 3D FE model cannot capture the boundary conditions and strain concentrations at the free edges precisely and efficiently. In addition, (Baroud et al. (2016)) proved the validity of the refined SCLS1 model for complicated examples. One we must highlight that even though the refined version LS1 model does not observe the same solution of the exact 3D boundary conditions at the free edges, it predicts the stress concentration with good approximation (8%).

1.8 Finite element discretization of the SCLS1 model

In the last years, an in-house finite element code named MPFEAP (MultiParticle Finite Element Analysis Program) was developed in Navier Laboratory. This code is committed to find finite element solution of the LS1 model. A standard Fortan 77 was used to write this code. In addition, this code is a development of the program MEF presented in (Dhatt and Touzot (1984)).
A new version of MPFEAP code was later on developed by (Baroud et al. (2016)) to get the finite element solution of SCLS1 model by adding a supplementary degree of freedom per interface. Baroud used a quadrilateral geometry of the finite element mesh defined in the (ξ, η) space as shown in the figure 1.4 :

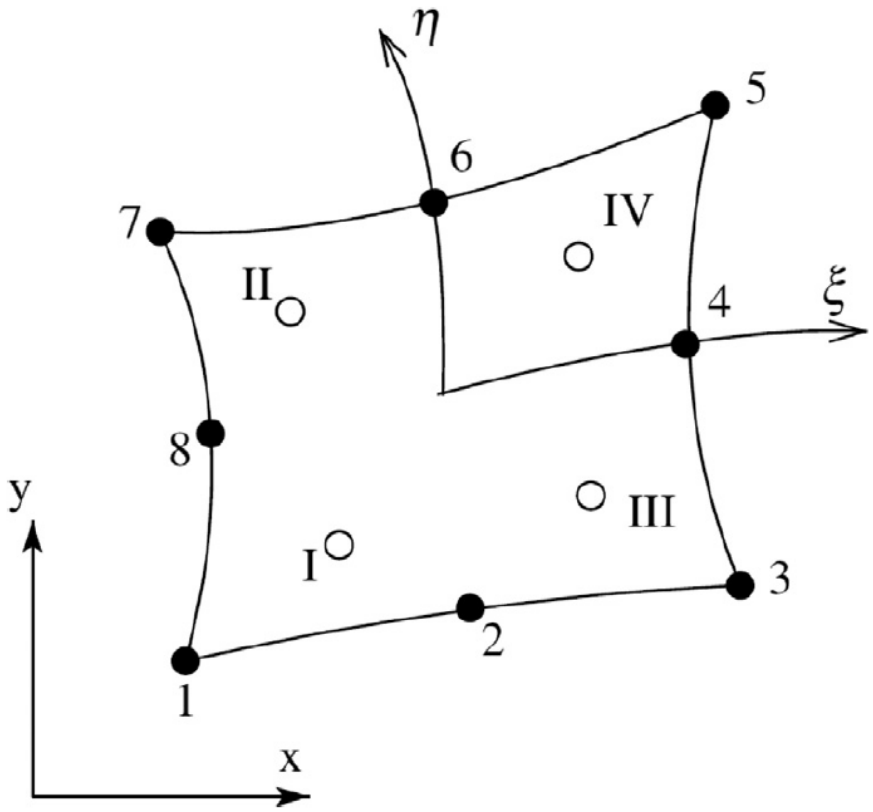


FIGURE 1.4 – The finite element-Quadrilateral element presented by (Baroud et al. (2016))

The interpolation geometry was written as :

$$\mathrm{x}(\xi,\eta) = \sum_{i=1}^{8} N_i(\xi,\eta)\mathrm{x}_i$$
$$\mathrm{y}(\xi,\eta) = \sum_{i=1}^{8} N_i(\xi,\eta)\mathrm{y}_i$$

where, $(\mathrm{x}_i, \mathrm{y}_i)$ are the coordinates of node i, $i = 1, \cdots, 8$ and $N_i(\xi,\eta)$ is the shape functions for $-1 \leq \xi \leq 1,\ -1 \leq \eta \leq 1$.

To modelise the SCLS1 model by finite element method, a finite-element discretization of the SCLS1 model has been proposed by (Baroud et al. (2016)) using the MPFEAP in-house software described in (Nguyen and Caron (2006)). In our numerical study, the SCLS1 multilayered plate model has been implemented in the open-source finite element FEniCS package (Alnaes et al. (2015) ;Logg et al. (2012b)). The FEniCs Project is a collection of free and open-source software components with the common goal to enable

automated solutions of differential equations. The components provide scientific computing tools for working with computational meshes, finite element variational formulations of ordinary and partial differential equations, and numerical linear algebra. We therefore benefit from FEniCS high-level domain-specific language for implementing the variational formulation associated with the SCLS1 model. Building upon the FEniCS implementation of a Reissner-Mindlin plate model (Bleyer (2018)), we define a generalized function space for the SCLS1 generalized displacement degrees of freedom.

More precisely, the retained discretization is based on a mesh of triangular elements with quadratic interpolation for all kinematical variables. As is the case for classical FE discretization of Reissner-Mindlin plate models, FE discretization of the SCLS1 model leads to shear-locking in the thin plate limit. Selective reduced integration is then used on the shear part of the strain (Bleyer (2018)).

In (Baroud et al. (2016)), the SCLS1 model has been compared with the LS1 model and reference 3D computations. This work showed that the SCLS1 model is as accurate as the LS1 model and is even closer to refined 3D solutions near free edges since it can correctly satisfy stress free boundary conditions. Besides, an intensive comparison between the LS1 model and other layerwise models derived from the Carrera Unified Formulation (CUF) family has been performed in (Thai et al. (2013)). The main conclusion of this work was that the LS1 model exhibits a similar accuracy to LM4 (mixed fourth-order) and LD3 (displacement third-order) layerwise models. This conclusion therefore also holds for the SCLS1 model considered here. Moreover, in contrast with these models, LS1 and SCLS1 exhibit much fewer degrees of freedom per node. For instance for a laminate with n = 4, LS1 has 20 (5n) dofs/node, SCLS1 has 23 (6n-1) whereas LD3 has 39 and LM4 has 102. One important feature of LS1 and SCLS1 is that no assumption is made on the displacement variations through the thickness but rather on the stress. Therefore, obtaining a complete 3D displacement field must be performed by a post-processing procedure which we will describe in the following chapter.

Chapitre 2

Mesh-adapted stress analysis of multilayered plates using a layerwise model

2.1 Résumé

Ce chapitre vise à exploiter une nouvelle modélisation par éléments finis et proposer une stratégie de remaillage pour les modèles multicouches. On présente tout d'abord la formulation des champs de déplacements reconstruits. On passe ensuite à présenter la stratégie de remaillage et les résultats numériques. Les résultats de ce chapitre ont fait l'objet d'un article publié dans le journal **Advanced Modeling and Simulation in Engineering Sciences** sous la référence (Salha et al. (2020)).

2.2 Abstract

This paper proposes a new finite-element modeling of a recent layerwise model for multilayered plates. This layerwise model is built from a specific 3D stress-field expansion along the thickness direction and involves, in particular, interlaminar transverse shear and out-of-plane stresses as generalized stresses. Its main feature is that 3D equilibrium equations and free-edge boundary conditions are directly taken into account into the stress-based construction of the model. A dual displacement-based finite-element discretization is implemented using the FEniCS software package and a remeshing strategy is proposed based on a novel error indicator. The error indicator is built based on the 3D stress field directly deduced from the layerwise generalized stresses and compared to a reconstructed stress field based on the model generalized displacements. The proposed error indicator is shown to identify the most critical parts of a laminate structure associated with complex 3D stress fields such as boundaries or stress concentration/singularity regions (near free-edges or delamination fronts). Through the combination of thickness discretization and in-plane mesh refinement in regions of interest, the proposed framework therefore offers an attractive alternative to 3D solid finite elements for an accurate prediction of stress states in composite laminates.

2.3 Introduction

Multilayered plates have very interesting mechanical properties that make them widely used in aerospace, automotive, telecommunication structures and civil engineering. A multilayered plate is represented as a pile of homogenized anisotropic plies made of fiber-reinforced composites. However, the highly anisotropic and heterogeneous nature of such laminates, the prediction of their overall properties is a challenging task. Free-edge effects are a major challenge when designing and analyzing such plates. Differences in the elasticity of adjacent layers have been shown to generally lead to high interlaminar stresses close to free edges (Vekua (1985); Barbero and J.N. (1991); Carrera (1998); Leguillon (1999); Chue and Liu (2002); Mittelsteda and Becker (2005)). Many models were derived accurately capture these free-edge effects. Highly detailed three-dimensional (3D)

finite-element models are computationally expensive and will only result in accurate stress predictions for sufficiently refined meshes since they rely on displacement interpolations. Two-dimensional plate models have therefore been introduced in order to simplify these computations while trying to keep a sufficiently accurate description of local 3D stress fields.

Equivalent single layer (ESL) models represent the laminate as an equivalent homogeneous plate. Many ESL models based on higher order theories have been proposed in literature (Wang and Choi (1982); Reddy (1984); Cho and Parmerter (1993); Cecchi and Sab (2007); Lebée and Sab (2011a,b, 2012); Swaminathan and Ragounadin (2004); Nguyen et al. (2008)) and are usually derived using two main approaches : asymptotic approaches and axiomatic approaches. The first class derives the plate model from the full 3D formulation of the problem, assuming the thickness of the plate goes to zero and using asymptotic expansion in which the leading order leads to Kirchhoff-Love plate theory (Ciarlet and Destuynder (1979)). The second approach is based on assuming a priori 3D fields, and the plate theory is derived by integration through the thickness and variational tools (Touratier (1992); Altenbach (1998); Kienzler (2002)). Although ESL models can provide acceptable results for the laminate global response, they may lead to very inaccurate estimations of the local response especially near free-edges.

Layerwise models, in which each layer is considered as an independent plate, have therefore been proposed to improve the local stress representation (Barbero and J.N. (1991); S.Botello et al. (1999); Carrera (1998); Moorthya and Reddy (1998); Gaudenzi et al. (1995); Robbins and Reddy (1993)). Layerwise models have been proved to be a very good alternative to 3D models since interpolation choices along the z-direction take into account the specificities of the laminate. The interested reader can refer to (Carrera (2003, 2004)) for a general overview of such models.
Following the ideas of Pagano's model (Pagano and Pipes (1970)), a layerwise model named LS1 was developed in (Chabot (1997); Carreria et al. (2002); Dallot and Sab (2008); Diaz et al. (2002); Lerpiniere et al. (2014); Nguyen and Caron (2006); Saeedi et al. (2012a,b, 2013a,b)).

The laminate is seen in this model as an overlap between Reissner-Mindlin plates, which are associated with interfacial stresses considered as additional generalized stresses. The main distinction between LS1 model and other models is that LS1 is a stress-based approach and others are displacement or mixed stress/displacement approaches.
The LS1 model, however, has some conceptual disadvantages, for instance because it cannot exactly fulfill 3D stress-free boundary conditions. Second, the LS1 model derives from the mixed variation theory Hellinger-Reissner, therefore, as the number of mathmatical layers by each physical layer increases, no theoretical guarantee is given to converge to the 3D model. Generalizing upon the same ideas, a layerwise model called statically compatible (SCLS1), was introduced in (Baroud et al. (2016)) in which the divergence of the interlaminar transverse shear stresses is introduced as an additional generalized stress.

Doing so, the SCLS1 model produces 3D stress field satisfying the local 3D balance equations and boundary conditions provided that their 2D plate counterparts are satisfied. The model can be derived through the complementary energy minimal principle, that ensures that the refined version converges to the exact 3D model with the increasing number of mathematical layers per physical layer.

When using computer simulations of mechanical or physical phenomena, numericals errors are very important to estimate in order to assess the quality of a solution. The discretization error induced by the finite-element method corresponds to the difference between the exact and approximation solutions. There are many methods for estimating the discretization error.
The first works Aziz (1972) ; Ciarlet (1978) proposed an estimation method called a priori estimation that is based on the data of the initial problem, especially their regularity, and on a mathematical analysis of the underlying problem. As a result, these estimators give good information on the convergences of the approached solutions but can only give a rough approximation of the error.
Then, a posteriori methods have been developed, which rely on some post-processing operation performed on the computed solution to estimate the error and thus allow a more accurate and reliable estimate. The a posteriori methods can be classified into three main families.
The first family of a posteriori methods is introduced by Zienkiewicz and Zhu (1987). These methods are called smoothing methods based on the defaults of regularities. The second family is called Residual-type error estimator proposed by Babuska and Rheinboldt (1987). These methods are based on the equilibrium defaults of the computed solution. In addition, these Residual methods are divided into two categories : explicit and implicit estimators. The implicit estimator is more expensive than the explicit estimator but gives more accurate results. In our study, we are interested in the third family of a posterioiri methods. This family has been introduced by Ladevéze (1975) consisting on reconstructing a statically admissible solution based on the kinematic one and evaluate the constitutive relation error. The major advantage of the error estimators based on the constitutive relation is that guaranteed upper bounds to the true error are obtained. Ainsworth and Oden (2000) presented a reasonable summary of a posteriori error estimate methods. Moreover, Sauter and Schwab (2011) developed and analyzed a posteriori error estimators.

Mesh generation is widely used in many engineering fields including those related to physical models described by partial differential equations (PDE). The first step in the simulations is to construct a mesh for the computational domain. In general, mesh construction for numerical simulation purposes involves many different fields and domains. Many authors (Frey and George (2008) ; Nielsen et al. (2013)) gives a fluent description for meshing and re-meshing structures.
Refinement procedures attempt at improving a solution accuracy by iteratively refining the mesh in regions of interest based on an error indicator. First, a solution with a spe-

cific mesh is obtained and an error is estimated. Second, mesh refinement is performed. Finally, a new solution is obtained and a new error computed, followed by a returning to second stage if this does not meet the requirements. We have many adaptation remeshing methods such as the h, r, p, hp-remeshing methods. The h-adaptation remeshing methods are the most useful, and the mesh is refined by increasing the number of degrees of freedom. Several possibilities of remeshing is possible in the h-remeshing method, first, the adaptation from parent mesh, where from an initial mesh the adapted mesh is built by subdividing the elements where the error is important by adding nodes. Second, adaptation with creation of a new mesh, where it is a complete remeshing of the domain, by building a whole new mesh with a refinement in the areas where the error is important and coarsening in the areas where the error is low. Finally, an adaptation with uniform refinement of all the mesh, in practicing this type it used to test the rate of convergence of the finite element method. Moreover, the r-adaptation remeshing method depends on improving the solution by moving the position of the nodes in the mesh to the areas where the error is important, without adding new ones and without modifying the connectivity. The p-adaptation remeshing method, depend on increasing the degree of interpolation of the elements while preserving the topology of the mesh. In addition, the combination of the h-remeshing and the p-remeshing methods give us the hp-adaptation remeshing method. The object of this method is to improve the precision of the calculation by taking advantage of the advantages of both methods. For example, we start with a refinement by method h to distribute the error uniformly, then we continue with a refinement by method p in order to increase the convergence rate.

Aiming at providing an operational tool for stress analysis in multilayered plates, this paper is concerned with the development of a mesh adaptation strategy based on an error indicator built from the local 3D stress field and a reconstructed 3D displacement field.

This chapter is organized as follows : the SCLS1 model finite-element implementation is discussed in section 1.8. Section 2.4.1 is dedicated the reconstruction of the 3D displacement and to error indicator computation used in the mesh adaptation. Finally, section 2.5 illustrate the method efficiency in capturing regions of interest in various configurations.

2.4 Mesh adaptivity based on field reconstructions

The SCLS1 model is quite expensive due to its high number of degrees of freedom per node. It can be seen as specific, mechanically-based, discretization in the z direction and can therefore be compared to a 3D discretization with a more accurate representation of the stress fields in the z direction. It becomes therefore beneficial to optimize the in-plane mesh for improved computational efficiency. The purpose of this section is to fulfil this goal by building an error indicator for mesh adaptation.

We propose to define this indicator as follows : from the finite-element computed generalized displacements $(U^i_\alpha, U^i_3, \Phi^i_\alpha, V^{j,j+1})$ fields in the (x, y)-plane, we first aim at recons-

tructing a 3D displacement field $\widehat{u}(x,y,z)$. We then derive the associated 3D strain and stresses using the local constitutive equation. The so-obtained reconstructed stress field $\widehat{\sigma}$ is then compared to the initial 3D stress $\boldsymbol{\sigma}^{3D}$ obtained from the generalized stresses $(N^i_{\alpha\beta}, M^i_{\alpha\beta}, Q^i_\alpha, \tau^{j,j+1}_\alpha, \nu^{j,j+1}, \pi^{j,j+1})$ via equations (1.7)-(1.9). See the illustration of the scheme in Figure 2.1.

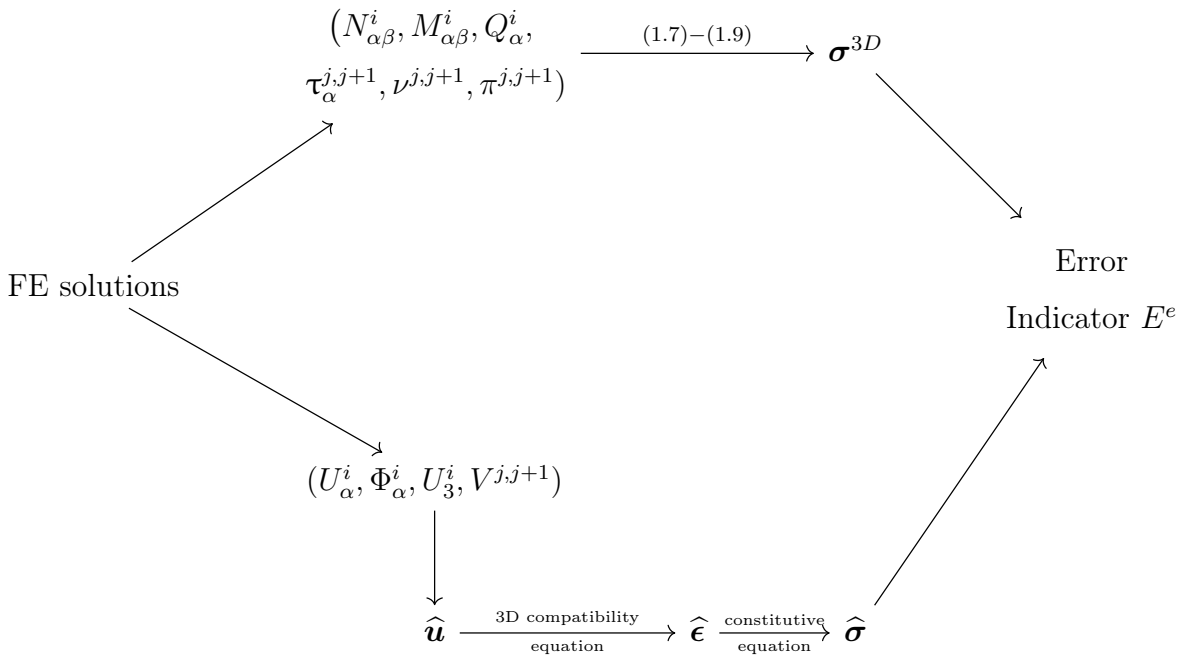


FIGURE 2.1 – The reconstruction scheme

2.4.1 Field reconstructions

In this section, we propose to reconstruct $\widehat{u}$ by considering a continuous piecewise linear variation of its components $\widehat{u}_i$ along the z direction. This interpolation will have to be as close as possible to satisfying equations (1.15)-(1.19). Let us mention that we tried other interpolations (in particular of higher-order) or reconstruction strategies but the latter gave the most satisfying results.

Let us first consider the in-plane displacement field. We first build an auxiliary in-plane displacement $\left(u^d_\alpha\right)$, with $\alpha = 1, 2$, as follows :

$$u^d_\alpha(x,y,z) = e^i \Phi^i_\alpha(x,y) P_1(z) + U^i_\alpha(x,y), \text{ for } z \in [h_i^-, h_i^+], \text{ and } i = 1, \cdots, n. \tag{2.1}$$

u^d_α is piecewise-linear and complies with equations (1.15),(1.16) but is not continuous at the interfaces. To achieve our initial goal, we rebuild from u^d_α the continuous displacement

field $\widehat{u}_\alpha$ by performing an L2-projection using the Least Square Method which can be stated as follows :
Find $[q_\alpha]$, with ${}^t[q_\alpha] = (q_\alpha^{1,2}, \cdots, q_\alpha^{n-1,n})$, that minimise

$$\int_{h_1^-}^{h_n^+} \left(\widehat{u}_\alpha(x,y,z) - u_\alpha^d(x,y,z)\right)^2 dz,$$

where,

$$\begin{aligned}
\widehat{u}_\alpha(x,y,z) &= \sum_{j=0}^{n} q_\alpha^{j,j+1}(x,y)\varphi^{j,j+1}(z),\\
q_\alpha^{0,1}(x,y) &= U_\alpha^1(x,y) - \frac{e^1}{2}\Phi_\alpha^1(x,y),\\
q_\alpha^{n,n+1}(x,y) &= U_\alpha^n(x,y) + \frac{e^n}{2}\Phi_\alpha^n(x,y).
\end{aligned}$$

Here, $q_\alpha^{0,1}$ and $q_\alpha^{n,n+1}$ are the displacements of the lower and upper faces of the plates, respectively, and $\varphi^{j,j+1}$ are the following basis functions :

$$\varphi^{j,j+1}(z) = \begin{cases} \dfrac{z - h_j^-}{h_j^+ - h_j^-} & \text{if } z \in [h_j^-, h_j^+] \\ \dfrac{z - h_{j+1}^+}{h_{j+1}^- - h_{j+1}^+} & \text{if } z \in [h_{j+1}^-, h_{j+1}^+] \quad \text{for } j = 1, \cdots, n-1 \\ 0 & \text{else} \end{cases}$$

and,

$$\varphi^{0,1}(z) = \begin{cases} \dfrac{z - h_1^+}{h_1^- - h_1^+} & \text{if } z \in [h_1^-, h_1^+] \\ 0 & \text{else} \end{cases}$$

$$\varphi^{n,n+1}(z) = \begin{cases} \dfrac{z - h_n^-}{h_n^+ - h_n^-} & \text{if } z \in [h_n^-, h_n^+] \\ 0 & \text{else} \end{cases}$$

The previous problem is equivalent to solving the following system :

Find $[q_\alpha]$, such that,

$$[A][q_\alpha] = [F^\alpha] \quad \text{for } \alpha = 1, 2,$$

with,

$$A_{ij} = \int_{h_1^-}^{h_n^+} \varphi^{i,i+1}(z)\varphi^{j,j+1}(z)dz, \text{ for } i,j = 1,\cdots,n-1,$$

and,

$$F_j^\alpha = \int_{h_1^-}^{h_n^+} \left(u_\alpha^d(x,y,z) - q_\alpha^{0,1}(x,y)\varphi^{0,1}(z) - q_\alpha^{n,n+1}(x,y)\varphi^{n,n+1}(z)\right)\varphi^{j,j+1}(z)dz,$$

for $j = 1,\cdots,n-1$.

Now, we aim to find the reconstructed out-of-plane displacement $\widehat{u}_3$ as a continuous piecewise linear function of z which is compatible with the generalized displacements U_3^i and $V^{j,j+1}$ in the sense of the following equations :

$$\left\{\begin{array}{l} \displaystyle\int_{h_i^-}^{h_i^+} \left(\frac{P_0^i(z)+P_2^i(z)}{e^i}\right)\widehat{u}_3(x,y,z) = U_3^i(x,y), \\ \displaystyle\int_{h_i^-}^{h_i^+} \left(P_1^i(z) - \frac{P_2^i(z)}{2}\right)\widehat{u}_3(x,y,z), \\ \displaystyle -\int_{h_{i+1}^-}^{h_{i+1}^+} \left(P_1^{i+1}(z) + \frac{P_2^{i+1}(z)}{2}\right)\widehat{u}_3(x,y,z) = V^{i,i+1}(x,y). \end{array}\right. \tag{2.2}$$

Introducing a continuous piecewise linear interpolation for $\widehat{u}_3(x,y,z)$ of the following form :

$$\widehat{u}_3(x,y,z) = \sum_{j=0}^{n} \tilde{q}^{j,j+1}(x,y)\varphi^{j,j+1}(z), \tag{2.3}$$

where $\varphi^{j,j+1}$ are linear shape functions and $\tilde{q}^{j,j+1}$ are the corresponding nodal values, the above equations become :

$$[B][\tilde{q}] = [F^3], \tag{2.4}$$

where ${}^t[\tilde{q}] = (\tilde{q}^{0,1},\cdots,\tilde{q}^{n,n+1})$, $[B]$ is a matrix of dimension $(2n-1, n+1)$ and $[F^3]$ a vector of dimension $(2n-1)$. The solution to the above problem is computed in the least-squares sense and gives a direct characterization of the degrees of freedom $[\tilde{q}]$ as a function of the generalized displacements U_3^i and $V^{i,i+1}$.

Finally, from the previously reconstructed 3D displacement field $\widehat{u}_i$, the strain field $\widehat{\boldsymbol{\epsilon}}$ is computed using the 3D compatibility equations and then the reconstructed stress tensor $\widehat{\boldsymbol{\sigma}}$ is computed using the 3D constitutive equations.

2.4.2 Error indicator and mesh adaptation

The error indicator which will be used for mesh adaptation is then computed based on the difference between $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ in terms of elastic energy. This error is computed for each triangular element :

$$E_e = \int_{\Omega_e} e\left(\boldsymbol{\sigma}^{3D} - \widehat{\boldsymbol{\sigma}}\right) d\Omega, \tag{2.5}$$

where $e(\boldsymbol{\sigma}) = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{S} : \boldsymbol{\sigma}$ and Ω_e denotes a given element e.

Each mesh element is then ordered in a decreasing fashion based on its error value : $E_1 > E_2 > \cdots > E_N$. where N is the total number of elements. We then tag the first K elements which contribute to at least a fraction η of the total error $E_{tot} = \sum_{e=1}^{N} E_e$:

$$\sum_{e=1}^{K} E_e > \eta E_{tot} \geq \sum_{e=1}^{K-1} E_e. \tag{2.6}$$

The tagged elements are then automatically refined by FEniCS mesh adaptation procedures. FEniCS refines the tagged elements automatically in the following way : The tagged elements are subdivided into 4 small triangles by adding new nodes on the triangle boundaries. Adjacent elements are not necessarily all tagged. In order to avoid hanging nodes in the untagged adjacent triangles, the latter are split into 2 smaller triangles.

2.5 Illustrative applications

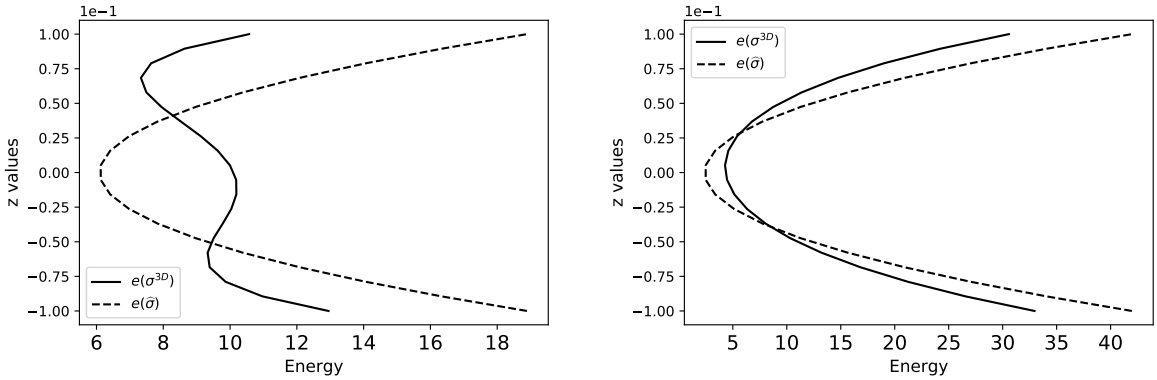
In this section, we investigate different illustrative applications assessing the quality of the stress field approximation, error indicator and mesh refinement strategy. The last examples consider more practical situations arising when designing composite laminates, namely stress concentrations near holes with associated free-edge singularities and interfacial stress singularities in the presence of interface delamination.

2.5.1 Homogeneous laminate

This first example considers a homogeneous square plate of length $l = 1$ and thickness $h = 0.2$. The constitutive material is assumed to be isotropic with $E = 10$ GPa and $\nu = 0.3$. The plate is fully clamped on its boundary and subject to a uniform loading of intensity $q = 8$. Calculations are performed considering a uniform discretization of $n = 1$, $n = 3$ and $n = 5$ layers across the thickness and have been compared to finite-element computations using 3D solid elements on a very fine mesh. The initial mesh was a structured mesh of with two triangular elements on each side of the square plate.

We start by the simple case with only one layer $n = 1$. The solution is of good quality

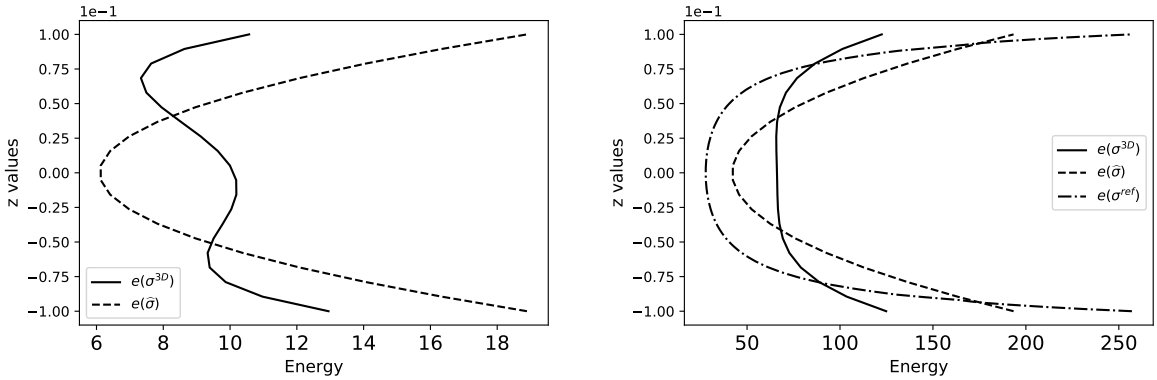
near the plate center after only one refinement step as shown in Figure 2.2



(a) Initial mesh (b) 1st mesh refinement

FIGURE 2.2 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the plate center ($n = 1$). (Energy in "GPa", z in "cm").

We now assess the quality of the computed stress field near the edge of the plate. In Figure 2.3 we plot the stress field near the left edge at the point $(x = 0.01, y = 0.5)$. And we compare the stress field $\boldsymbol{\sigma}^{3D}$ with its reconstruction at the same point near the left edge, that show that the reconstruction field doesn't agree with the 3D stress.This indicates that we should refine the plate in this region. In addition, Figure 2.3b shows that after 6 steps of refinement we get better agreement between the 3D stress field $\boldsymbol{\sigma}^{3D}$ and its reconstruction field $\widehat{\boldsymbol{\sigma}}$.



(a) Initial mesh (b) 6st mesh refinement

FIGURE 2.3 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the plate edge ($n = 1$). (Energy in "GPa", z in "cm").

The error indicator identifies the region placed near the edges as the most critical regions

as shown in the Figure 2.4 after 6 refinement steps.

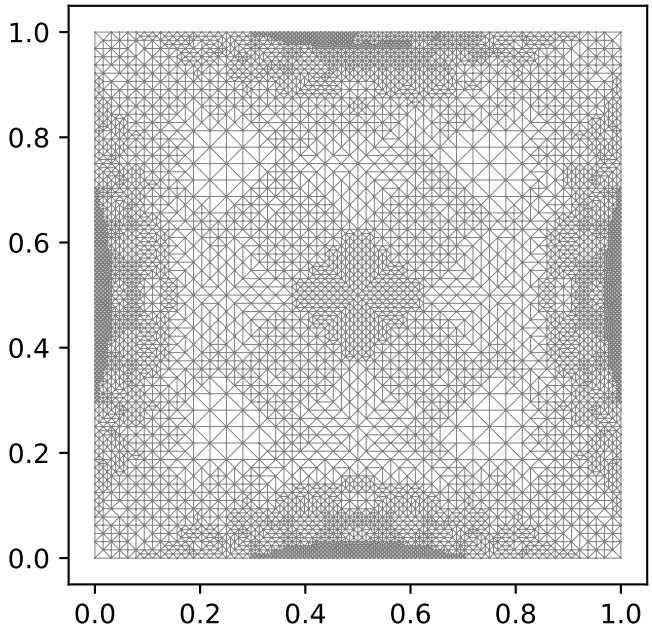
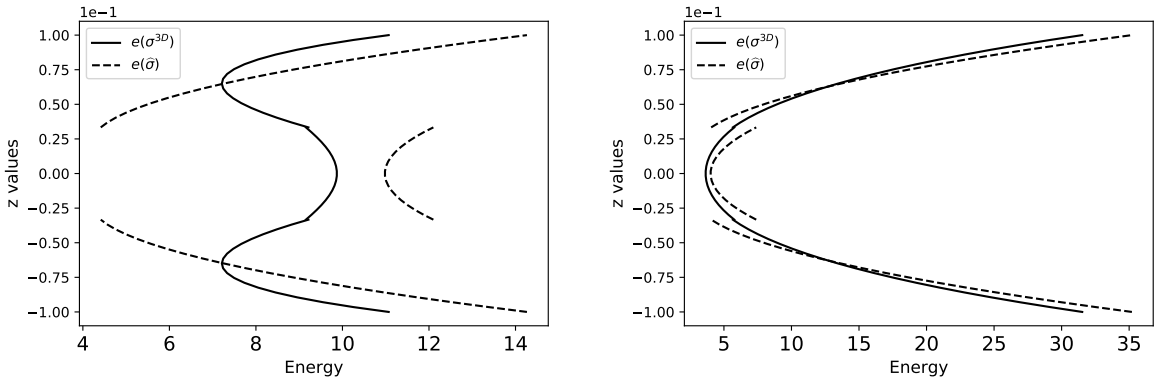


FIGURE 2.4 – Final refined meshes for the homogeneous plate for $n = 1$

Then, the case with $n = 3$ layers is considered. As expected, the multilayered plate solution is of very good quality near the plate center after only one refinement step as shown in Figure 2.5.



(a) Initial mesh

(b) 1st mesh refinement

FIGURE 2.5 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the plate center ($n = 3$). (Energy in "GPa", z in "cm").

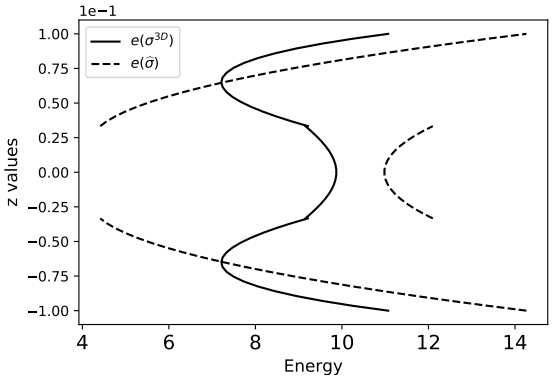
We therefore investigate the quality of the computed stress field at a point of coordinate $(x = 0.01, y = 0.5)$ near the left edge. In Figure 2.6a, we compare the multilayered stress field $\boldsymbol{\sigma}^{3D}$ with its reconstruction as described in section 2.4.1 at the same point near the edge. It can be observed that the reconstruction does not agree with $\boldsymbol{\sigma}^{3D}$ for the initial coarse mesh, indicating that mesh size should be refined in this region. Figures 2.6b and 2.6c illustrate the evolution of $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ near the border when refining the mesh. It can be seen that mesh refinement provides a much better agreement between both stress fields. The error indicator therefore correctly identifies regions located near the clamped boundaries as the most critical regions as evidenced by the final mesh layout of Figure 2.7a obtained after 6 refinement steps.

Performing the same comparison in the case when the plate thickness is discretized in $n = 5$ layers shows the same behaviour (Figure 2.8). Although $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ are a little closer for the initial coarse mesh, the deviation is still significant indicating that in-plane mesh resolution is not fine enough. The mesh refinement procedure yields a similar final mesh layout, with fine cells concentrated along the borders (see Figure 2.7b), and better agreement between $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the final stage. On both Figures 2.6c and 2.8c, the reference solid FE solution is also represented, showing a good agreement with the multilayered stress field after mesh refinement.
More insight can also be gained when observing the error indicator per individual layer. For example, Figure 2.9, 2.10 and 2.11 plot the error contribution in the first, second, and third layer respectively before and after re-meshing. Note that, the errors in layers 4 and 5 are, respectively, the same as in layers 2 and 1 due to the symmetry of the problem with respect to the mid plane. In addition, it is remarkable that the error contribution is concentrated near the borders.
The effect of mesh refinement is further illustrated when plotting the evolution of the

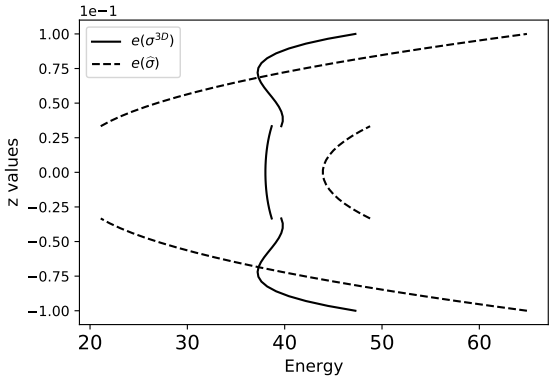
total relative error indicator in Figure 2.12, defined as :

$$E^r = \frac{E_{tot}}{S}, \text{ with } S = \int_\Omega e(\boldsymbol{\sigma}^{3D}) d\Omega.$$

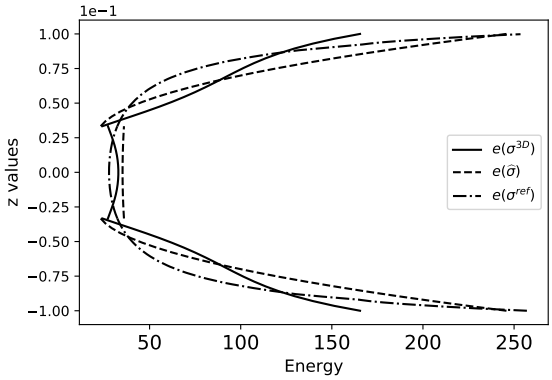
It can be seen that the relative errors decrease when refining the mesh and tend to stabilize after a few iterations only. Besides, relative errors are larger for $n = 3$ than $n = 5$ which may indicate that the mesh reconstruction if of higher quality for $n = 5$ layers than $n = 3$ layers. Let us point out that the value obtained for such errors cannot be considered neither as a guaranteed level of error with respect to an exact solution nor as an upper bound to the true error. It is however an error indicator, as showed by the previous results, which can be used qualitatively to assess the solution accuracy.



(a) Initial mesh

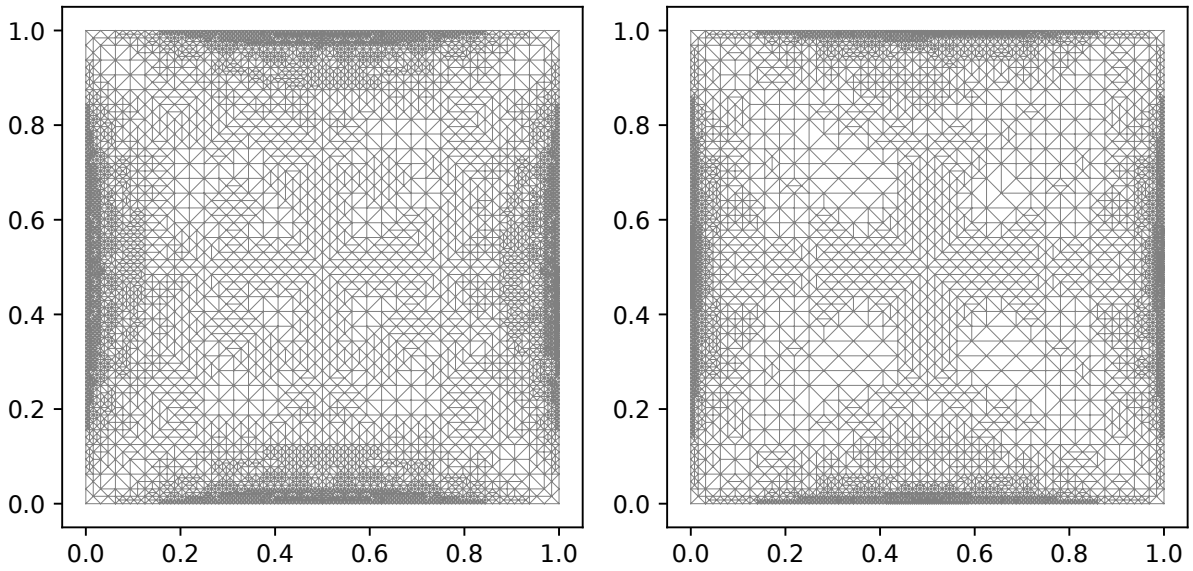


(b) 2nd mesh refinement



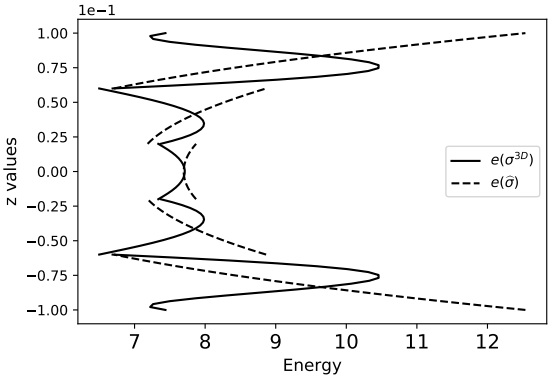
(c) 6th mesh refinement

FIGURE 2.6 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the plate edge for $n = 3$.(Energy in "GPa", z in "cm").

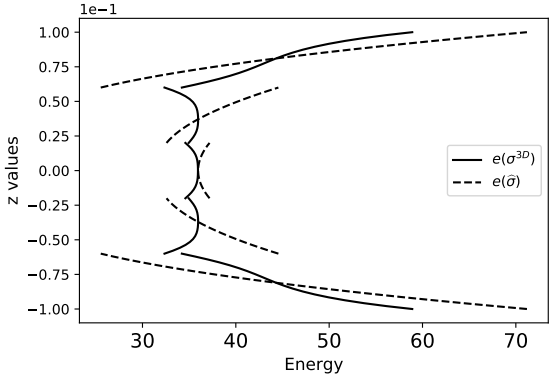


(a) $n = 3$ (b) $n = 5$

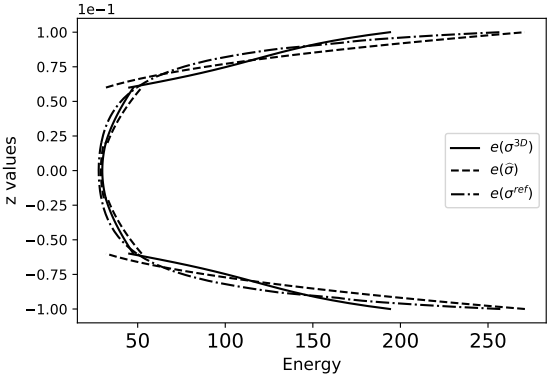
FIGURE 2.7 – Final refined meshes for the homogeneous plate for different thickness discretization levels



(a) Initial mesh

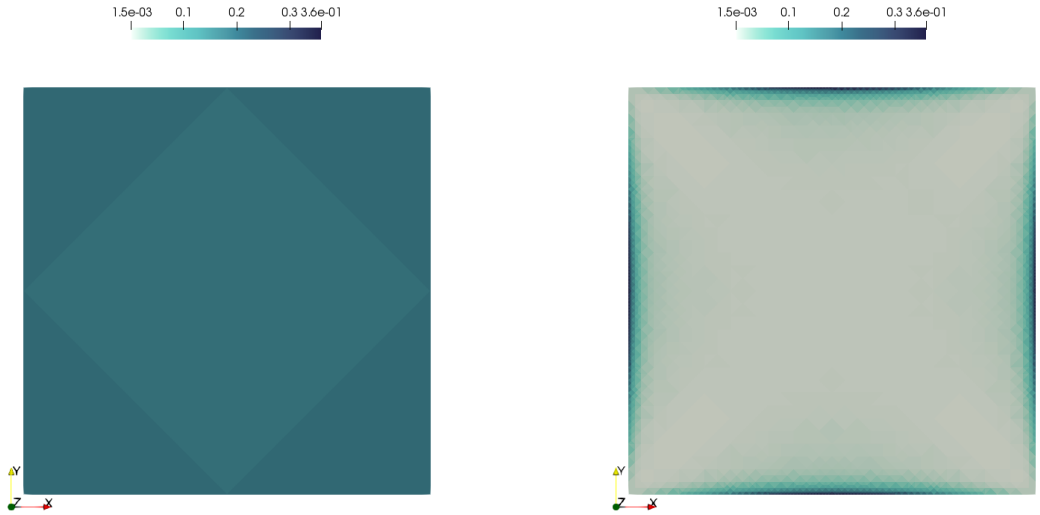


(b) 2^{nd} mesh refinement



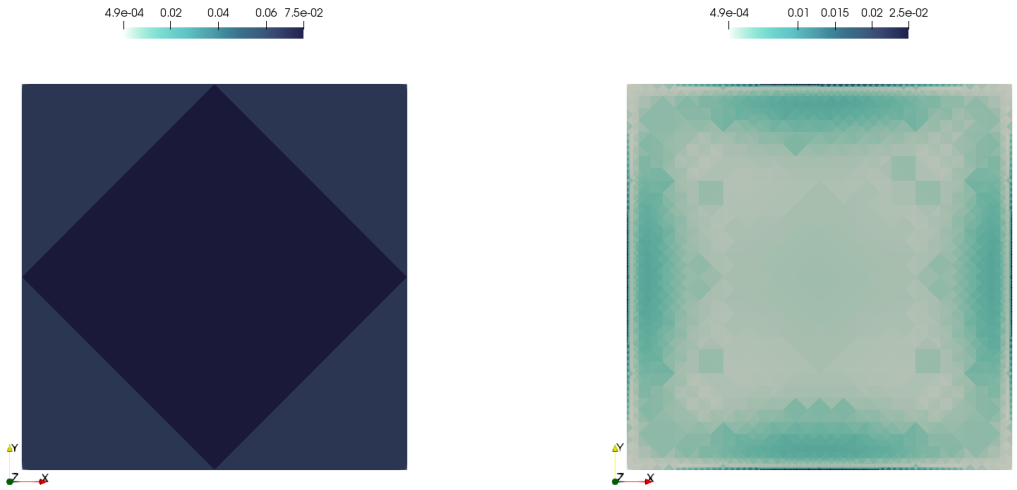
(c) 6^{th} mesh refinement

FIGURE 2.8 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$, $\widehat{\boldsymbol{\sigma}}$ and $\boldsymbol{\sigma}^{ref}$ at the plate edge for $n = 5$. (Energy in "GPa", z in "cm").



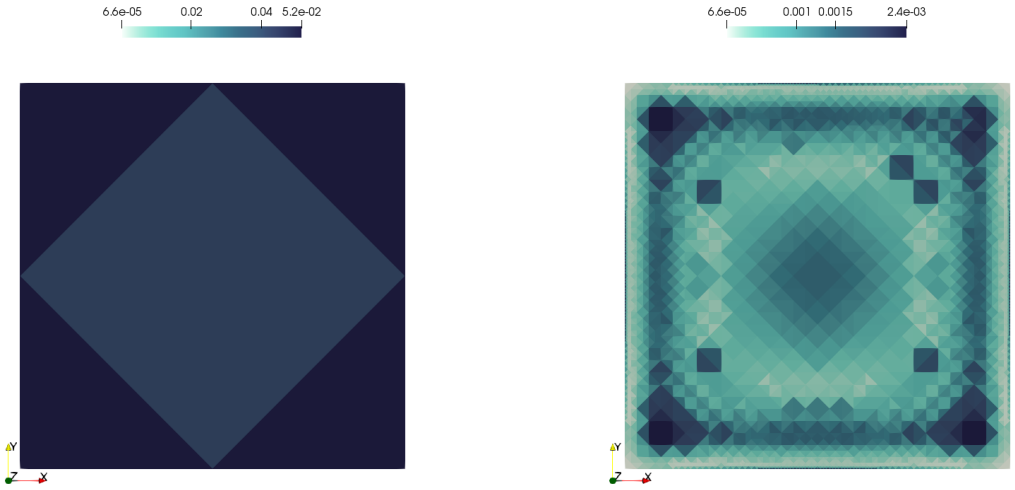
(a) Error in first layer for initial mesh (b) Error in first layer after re-meshing

FIGURE 2.9 – Error indicator maps in layer 1 for initial and re-meshing mesh.



(a) Second layer error for initial mesh (b) Second layer error after re-meshing

FIGURE 2.10 – Error indicator maps in layer 2 for initial and re-meshing mesh.



(a) Third layer error for initial mesh (b) Third layer error after re-meshing

FIGURE 2.11 – Error indicator maps in layer 3 for initial and re-meshing mesh.

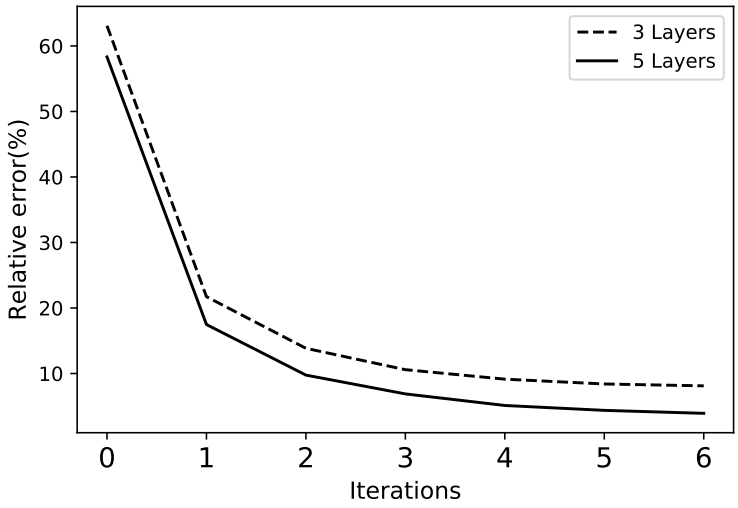


FIGURE 2.12 – Total relative error evolution for 3 and 5 layers discretizations

2.5.2 Triple laminate

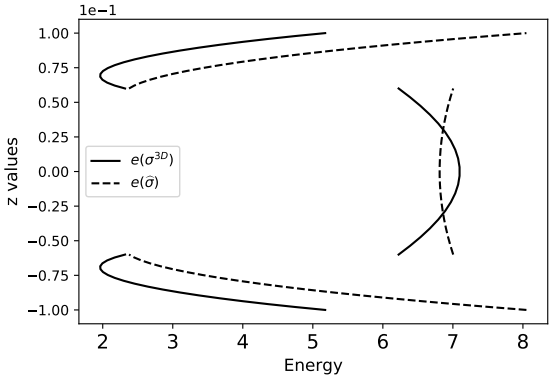
The second example considers a heterogeneous square plate of length $l = 1$ and total thickness $h = 0.2$ made of a triple laminate consisting of a central core of thickness $e_2 = 0.12$ and two symmetric skins of thickness $e_1 = 0.04$ each. The constitutive materials are assumed to be isotropic with $E = 50$ GPa and $\nu = 0.2$ for the skins and

$E = 10$ GPa and $\nu = 0.3$ for the core. Loading and boundary conditions are the same as for the homogeneous plate. Calculations are performed considering a discretization consisting of one mathematical layer in both skins and in the core (total of $n = 3$ layers) and a discretization consisting of one mathematical layer per skin and 3 layers for uniformly discretizing the core thickness (total of $n = 5$ layers). Again the multilayered plate model computations have been compared to reference 3D solid finite-element computations on a fine mesh.

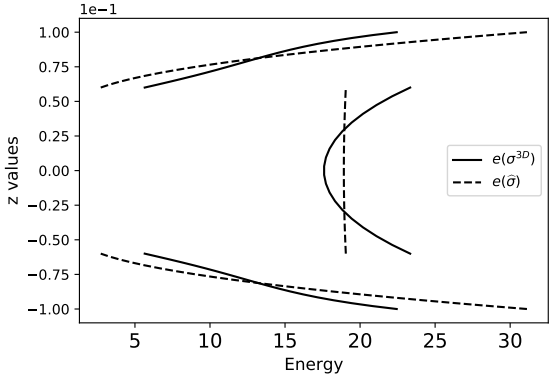
First, we considered the case with $n = 3$ layers. As before, the solution is of lesser quality near the supports and stress fields are therefore compared at the same $(x = 0.01, y = 0.5)$ location as before. Figure 2.13 shows the comparison of the stress field $\boldsymbol{\sigma}^{3D}$ with its reconstruction across the plate thickness for various mesh refinement steps. It can be observed that $\boldsymbol{\sigma}^{3D}$ and its reconstruction do not match for the initial coarse mesh, indicating that the mesh size should be refined in this region. Mesh adaptation improves the quality of the solution in such regions as evidenced by the good agreement with the reference 3D FE solution.

Performing the same comparison using a more refined discretization with $n = 5$ layers in the thickness exhibits a similar behaviour (Figure 2.14). Although $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ are a little closer for the initial coarse mesh, the deviation is still significant indicating that in-plane mesh resolution is not fine enough. A similar refined mesh layout is obtained with fine cells concentrated along the borders (see Figure 2.15), and better agreement between $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the final stage.

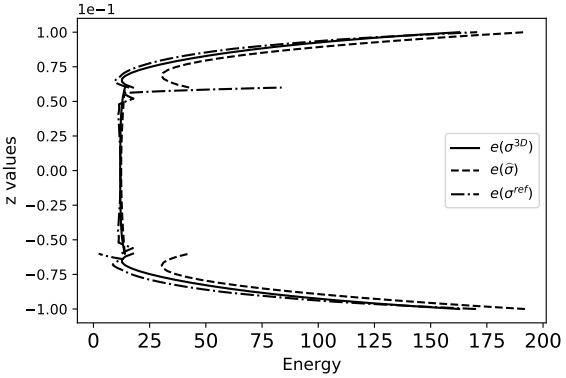
Finally, the evolution of the total relative error indicator as a function of mesh refinement steps in Figure 2.16 exhibits a similar behaviour as for the homogeneous plate.



(a) Initial mesh

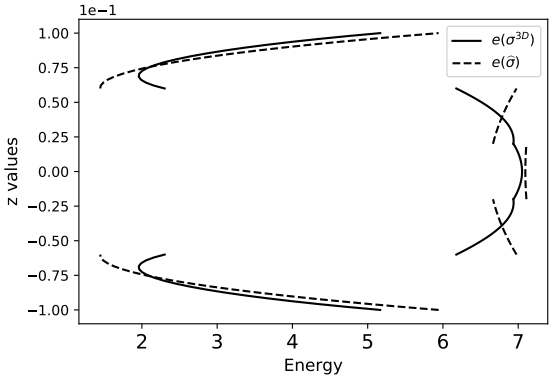


(b) 2^{nd} mesh refinement

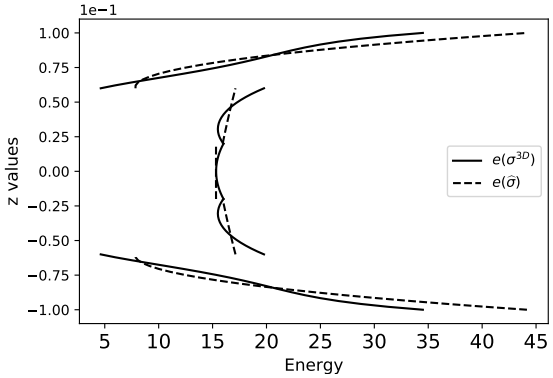


(c) 6^{th} mesh refinement

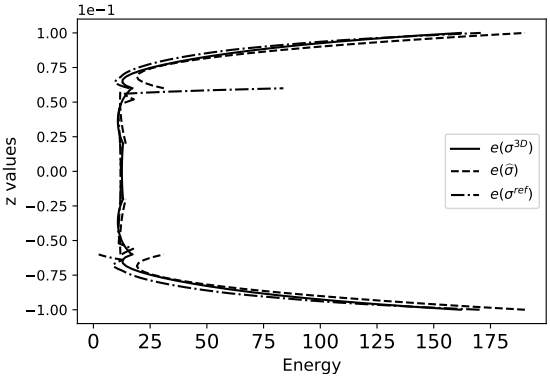
FIGURE 2.13 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$ and $\widehat{\boldsymbol{\sigma}}$ at the plate edge for $n = 3$.(Energy in "GPa", z in "cm").



(a) Initial mesh

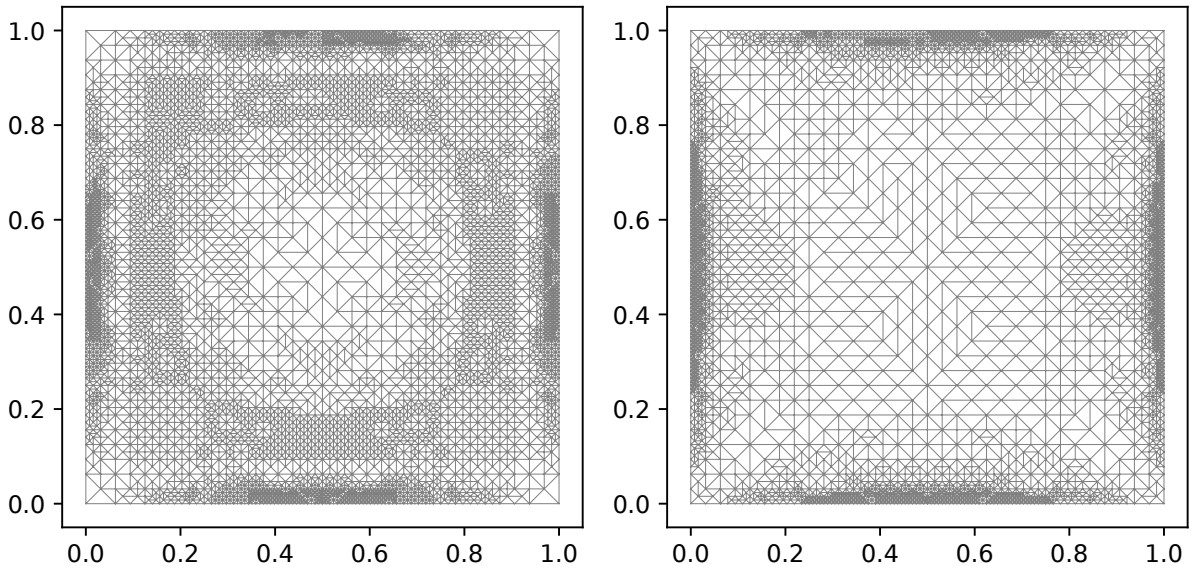


(b) 2^{nd} mesh refinement



(c) 6^{th} mesh refinement

FIGURE 2.14 – Energy densities across the plate thickness computed for $\boldsymbol{\sigma}^{3D}$, $\widehat{\boldsymbol{\sigma}}$ and $\boldsymbol{\sigma}^{ref}$ at the plate edge for $n = 5$. (Energy in "GPa", z in "cm").



(a) $n = 3$ (b) $n = 5$

FIGURE 2.15 – Refined meshes for the triple laminate for different thickness discretization levels

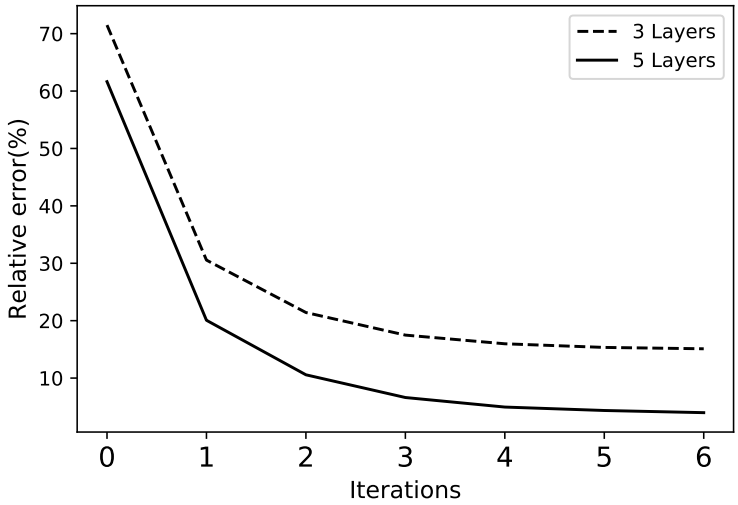
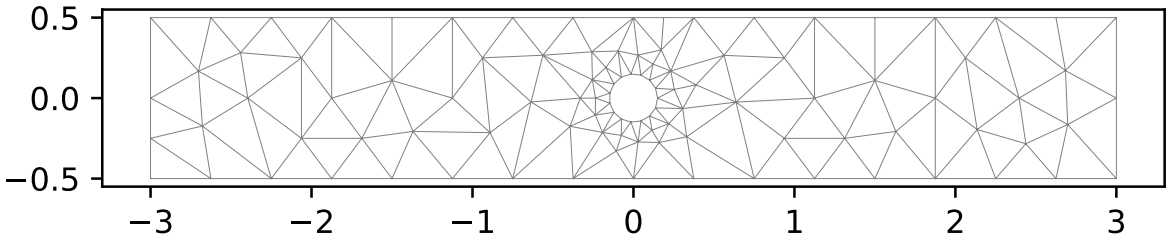


FIGURE 2.16 – Total relative error evolution for 3 and 5 layers discretizations

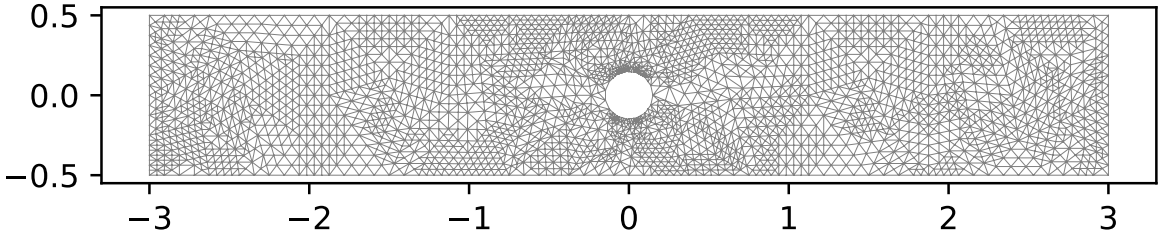
2.5.3 Laminate with a circular hole

The third example considers a rectangular multilayered plate of length $l = 6$, width $w = 1$ and total thickness $h = 0.01$. The plate is perforated by a circular hole of radius $R = 0.15$ in its center (Figure 2.17a). The laminate is made of a transversely isotropic material

of elastic properties $E_T = 14.48$ GPa, $E_L = 137.9$ GPa, $\nu_T = 0.21$, $\nu_L = 0.21$, $\mu_T = 5.86$ GPa and $\mu_L = 5.86$ GPa with L (resp. T) denoting the fiber longitudinal direction (resp. the perpendicular transverse direction). The laminate consists of 6 plies (one layer per ply) with fibers oriented at $[0°, 90°, 45°, -45°, 90°, 0°]$ with respect to the horizontal direction. A tensile loading is applied to the plate through an imposed horizontal displacement $\boldsymbol{U}^i = \pm \mathtt{U} \boldsymbol{e}_x$ for all plies $i = 1, \ldots, 6$.



(a) The initial mesh



(b) The final refined mesh

FIGURE 2.17 – Mesh refinement for the plate with a circular hole

Applying the proposed reconstruction and error estimation on this problem yields to a globally more refined mesh with finer regions located near the top and bottom boundaries of the circular hole, see Figure 2.17b obtained after 4 refinement steps.

More insight can also be gained at visualizing the individual layer contributions to the total error. For instance, Figure 2.18 plots the contribution of the 45° (layer 3) and −45° (layer 4) layers to the total error. These two contributions are the most dominant one as regards stress concentrations near the hole. The effect of the material anisotropy on these two contributions can also be clearly observed.

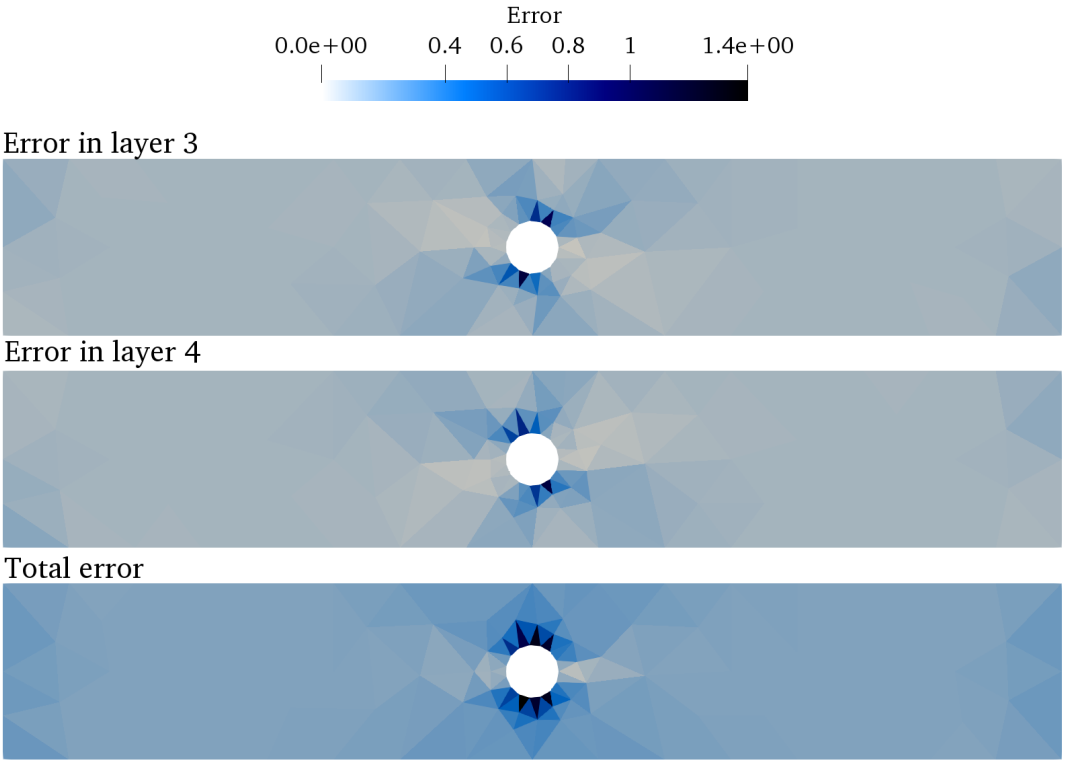


FIGURE 2.18 – Error indicator maps in layers 3 and 4 (top and middle) and total error for all layers (bottom) on the initial mesh.

2.5.4 Double-Cantilever Beam with delaminated interface

The final example we consider is that of a rectangular multilayered plate of the same dimensions as before (without the circular hole) and the same lamination properties. We model a portion of a delaminated interface located in the middle interface ($(i, i+1) = (3, 4)$) in the region $x \leq 1$ by forcing the interface stresses $\nu^{3,4}$ and $\tau_\alpha^{3,4}$ to be zero on this region. This results in an appropriate modification of the constitutive equations of the SCLS1 model and the corresponding finite-element implementation.

The plate is clamped on its right boundary, and positive (resp. negative) vertical displacements $U_3^i = +\mathtt{U}$ (resp. $U_3^i = -\mathtt{U}$) are enforced on the left part for the top layers $i = 4, 5, 6$ (resp. bottom layers $i = 1, 2, 3$), simulating a Double-Cantilever Beam test (see Figure 2.19).

As expected, the mesh adaptation procedure mainly concentrates the finer cells near the delamination front at which interface stresses are the most singular, see Figure 2.20. The proposed procedure can therefore be considered to be coupled with a delamination propagation model for which stresses driving the delamination front propagation will be well resolved.

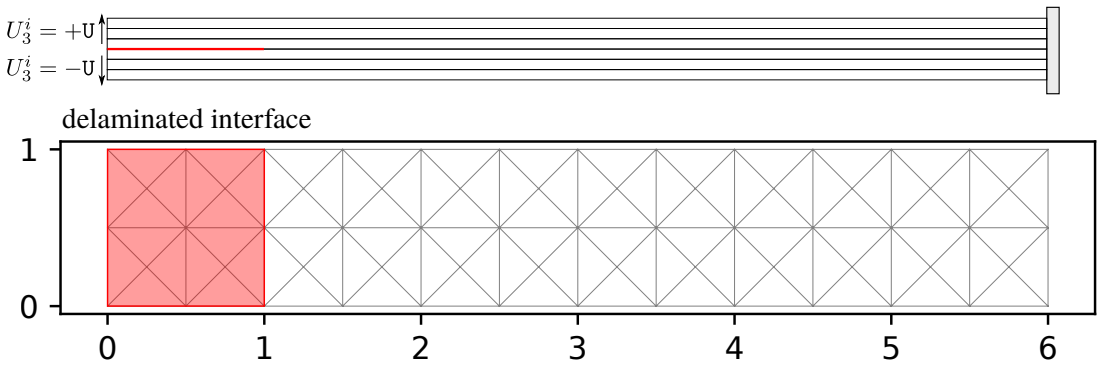


FIGURE 2.19 – The initial mesh for the DCB problem

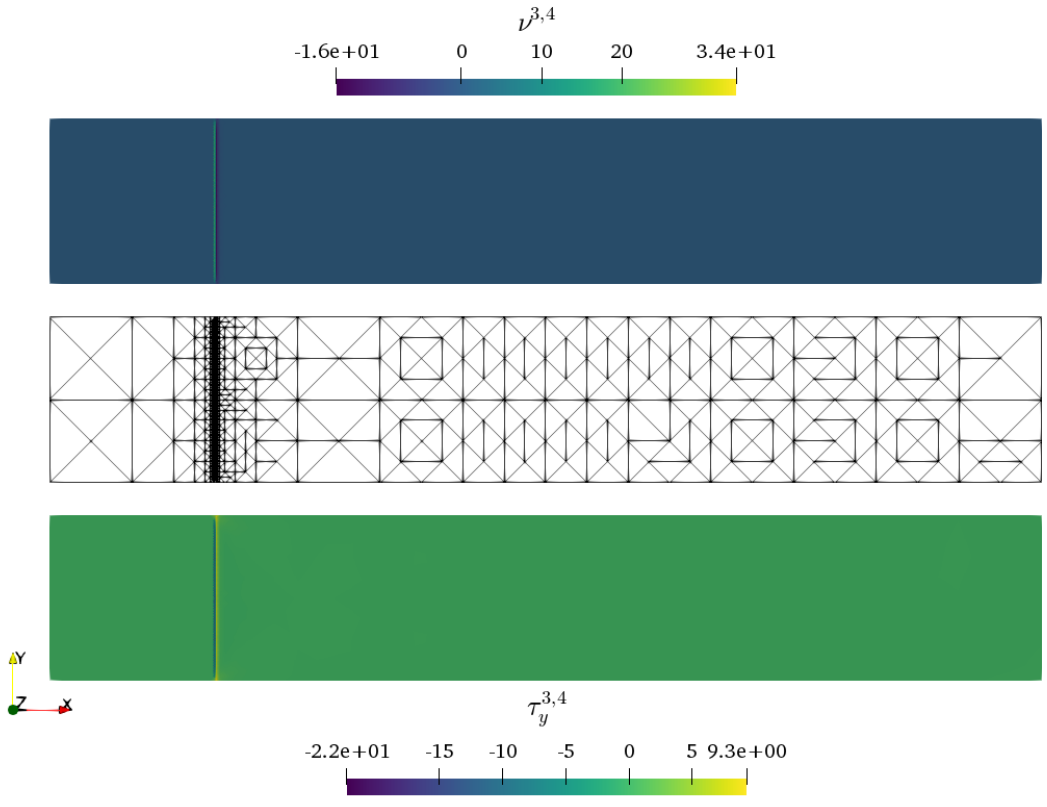


FIGURE 2.20 – Top : out-of-plane interface stress $\sigma_{zz}(z=0) = \nu^{3,4}$. Middle : final adapted mesh. Bottom : interface shear stress $\sigma_{yz}(z=0) = \tau_y^{3,4}$

2.6 Conclusions and perspectives

In this chapter, a statically compatible layerwise stress model for laminated plates (SCLS1) is considered for an accurate representation of 3D elastic fields. A mesh adaptation strategy is then developed which relies on the reconstruction of 3D displacement fields from the model generalized displacements, the error indicator being obtained by a constitutive error between both fields. Illustrative examples show that the method is indeed able to refine the mesh in regions with complex 3D stress fields such as straight edges, notches or delamination fronts. This adaptive method paves the way to further developments including interface delamination propagation or damage at the ply level.

The proposed methodology can be further improved by pointing out that refined layerwise models such as the one considered here is appropriate in critical regions near boundaries, free-edges, delaminated interfaces, etc. This point is indeed properly identified by the proposed remeshing procedure. In the bulk region away from these critical zones, it would we sufficient to adopt an equivalent single-layer plate model based on a Love-Kirchhoff kinematics for instance. Although the remeshing procedure favours coarse cells in such regions, mitigating the number of unnecessary degrees of freedom, an additional gain could then be obtained by mixing a layerwise model for critical regions with an equivalent single-layer model for the remaining part.
A second potential line of work is concerned with the fact that, although the layerwise model is built at the continuous level from a stress-based perspective complying with the balance equations, its numerical resolution is performed through a displacement-based approximation for the in-plane variations. As a consequence, the resulting generalized stress fields, and therefore, the associated 3D stress field, do not satisfy strongly the balance equations. In order to maintain the initial philosophy of a stress-based statically compatible construction, developing a stress-based finite-element discretization of the model would be an interesting approach, potentially paving the way to obtain more rigorous error estimators than the one considered here.

Chapitre 3

A hybridized mixed approach for efficient stress prediction in a layerwise plate model

3.1 Résumé

Ce chapitre vise à exploiter la méthode d'hybridation mixte pour le modèle SCLS1. On présente tout d'abord la méthode hybride mixte pour le modèle 3D. Puis on passe à la méthode d'hybridation mixte pour le modèle SCLS1. A la fin, on présente les résultats illustartifs de cette méthode. Les résultats de ce chapitre ont fait l'objet d'un article soumis dans un journal scientifique.

3.2 Abstract

Building upon recent works devoted to the development of a stress-based layerwise model for multilayered plates, we explore an alternative finite-element discretization to the conventional displacement-based finite-element method. We rely on a mixed finite-element approach where both stresses and displacements are interpolated. Since conforming stress-based finite-elements ensuring traction continuity are difficult to construct, we consider a hybridization strategy in which traction continuity is relaxed by the introduction of an additional displacement-like Lagrange multiplier defined on the element facets. Such a strategy offers the advantage of uncoupling many degrees of freedom so that static condensation can be performed at the element level, yielding a much smaller final system to solve. Illustrative applications demonstrate that the proposed mixed approach is free from any shear-locking in the thin plate limit and is more accurate than a displacement approach for the same number of degrees of freedom. As a result, this method can be used to capture efficiently strong intra- and inter-laminar stress variations near free-edges or cracks.

3.3 Introduction

Multilayered plates are important in structural engineering and are widely studied by engineers during the 20^{th} century with applications ranging from aerospace engineering to civil engineering. The materials in each layer can be either homogeneous and isotropic or heterogeneous and anisotropic (e.g. fiber-reinforced composites). The difficulty in studying such structures comes from the strong variations of mechanical properties between each ply, especially when using anisotropic materials. One of the major issues in design and analysis of such multilayered plates is related to free-edge effects. In fact, near free edges there are highly concentrated interlaminar stresses (Ting and Chou (1981); Wang and Choi (1982); Leguillon (1999); Chue and Liu (2002); Mittelsteda and Becker (2005)). Such stress concentrations are important to account for since they are at the origin of interlayer delamination and failure of the laminate. Many models have been derived to properly describe such free-edges effects. Highly detailed three-dimensional (3D) finite-element models are expensive and will result in accurate stress predictions only for sufficiently refined meshes since they rely on displacement interpolations. Two-dimensional

plates models attempts at simplifying these computations while trying to keep a sufficiently correct description of local 3D stress fields.
Many equivalent single layer (ESL) models that represent the laminate as an equivalent homogeneous plate are proposed in literature and are based on higher order theories (Whitney (1973); Reddy (1984); Cho and Parmerter (1993); Cecchi and Sab (2007); Lebée and Sab (2011a,b, 2012); Swaminathan and Ragounadin (2004); Vidal and Polit (2008, 2011)). Although ESL models can provide acceptable results for the laminate global response, they may lead to very inaccurate estimations of local response especially near free-edges.

Conversely, layerwise models, in which each layer is considered as an independent plate, have therefore been proposed to improve the local stress representation (Barbero and J.N. (1991); Robbins and Reddy (1993); Gaudenzi et al. (1995); Moorthya and Reddy (1998); S.Botello et al. (1999); Carrera (1998)). They have been proved to be a very good alternative to 3D models when using proper interpolation along the thickness direction to take into account the laminate material distribution. The interested reader can refer to (Carrera (2002, 2004); Zhang and Yang (2009)) for a general overview on such models.

Following Pagano's model (Pagano and Pipes (1970)), a layerwise model named LS1 was developed in (Chabot (1997); Carreria et al. (2002); Diaz et al. (2002); Lerpiniere et al. (2014); Nguyen and Caron (2006); Saeedi et al. (2012a,b, 2013a,b)). In this model, the laminate is considered as a superposition of Reissner-Mindlin plates linked together by interfacial stresses which are considered as additional generalized stresses. A layerwise model called statically compatible (SCLS1), has been later introduced in (Baroud et al. (2016)) in order to generalize the LS1 model by a purely statically compatible construction. Doing so, the SCLS1 model produces a 3D stress field satisfying the local 3D balance equations and boundary conditions provided that their 2D plate counterparts are satisfied. Moreover, this model is derived by means of the minimum of the complementary potential energy ensuring the convergence of its refined version to the exact 3D model, as the number of mathematical layers per physical layer increases.

Such models have generally been solved numerically using a classical displacement-based finite-element approach, as in (Nguyen and Caron (2006); Baroud et al. (2016)). A notable exception is the recent implementation of the LS1 model of (Nasser et al. (2018b)) which relies on a mixed finite-element approach. Mixed finite-element approaches are appealing since stress quantities are also interpolated, as opposed to pure displacement-based approaches in which they are post-processed from the displacement solution. This interpolation usually results in higher quality of the stress fields which are the principal quantity of interest in engineering applications. Mixed approaches are however more difficult to implement ; they yield a saddle point problem, and result in a much higher system dimension. This difficulty may well render mixed methods less efficient than displacement-based methods using a much finer mesh. In the present contribution, we will explore the use of hybridized mixed methods which not only offer similar advantages in terms of stress field

accuracy, but also result in smaller system size by using static condensation of the stress unknowns. We will show that these methods can be more accurate than a displacement approach for the same number of degrees of freedom. The methodology of hybridized mixed methods is general enough to be applied on a complex model as the SCLS1 layerwise model.

This chapter is organized as follows : In 3.4 we describe the finite-element displacement-based implementation. Hybridized mixed methods for 3D models are then reviewed in section 3.5. Section 3.6 is then devoted to the application of hybridized mixed methods to the SCLS1 model. Finally, section 3.7 is dedicated to numerical examples demonstrating the efficiency of the hybridized mixed approach.

3.4 Finite-element displacement-based implementation

The numerical study of the SCLS1 multilayered plate model has been proposed by (Baroud et al. (2016)) using the MPFEAP in-house described in (Nguyen and Caron (2006)), and in (Salha et al. (2020)) using the open-source finite element FEniCS package (Logg et al. (2012b); Alnaes et al. (2015)). This implementation follows a classical displacement-based finite-element discretization using a continuous quadratic interpolation for all generalized kinematical variables $U^i_\alpha, U^i_3, \Phi^i_\alpha$ and $V^{j,j+1}$ on a triangular 2D mesh. As discussed in (Logg et al. (2012b); Baroud et al. (2016)), the finite-element discretization suffers from shear-locking issues in the thin plate limit which is alleviated using selective reduced integration.

3.5 Hybridized mixed methods for 3D continua

Mixed approaches for 3D continua consist in considering a simultaneous interpolation for the stress variable σ and the displacement u (Fraeijs de Veubeke (1965); Arnold and Brezzi (1985); Cockburn et al. (2009); Brezzi and Fortin (2012)). The interpolation space for σ is often chosen to satisfy the traction continuity condition. If choosing such spaces is feasible when u is a scalar (e.g. in the case of antiplane elasticity), this is much harder for the general vectorial case of 2D/3D elasticity (Arnold and Winther (2002); Arnold et al. (2007); Gong et al. (2019)). Hybridized mixed methods therefore consist in relaxing the a priori traction continuity requirement and including it in the variational formulation (Gibson et al. (2020)). They are therefore easier to formulate, especially regarding the choice of the stress interpolation space, and offer computational advantages as it will be seen in the following section.

3.5.1 Continuous variational formulation

Let us consider a domain Ω with imposed displacements $u_i = u_i^0$ on a Dirichlet part $\partial\Omega_D$ of the boundary and imposed tractions $\sigma_{ij}n_j = T_i^0$ on the remaining Neumann part $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$. Let us also denote by Γ the set of internal lines of stress discontinuities (typically inner edges of a finite-element mesh) and introduce the jump operator through Γ as follows : $[\![v]\!] = v^+ + v^-$ where $\pm$ are arbitrarily defined sides of Γ. n^+ (resp. n^-) will denote the unit normal of Γ pointing outwards of the $+$ (resp. $-$) side.

Let us start with the complementarity energy principle which states that the solution (in terms of stresses) minimizes the following complementarity energy under static equilibrium conditions :

$$\begin{aligned} \min_{\sigma} \quad & \int_\Omega \frac{1}{2}\sigma_{ij}S_{ijkl}\sigma_{kl}d\Omega - \int_{\partial\Omega_D} \sigma_{ij}n_j u_i^0 dS, \\ \text{s.t.} \quad & \sigma_{ij,j} + f_i = 0 \quad \text{in } \Omega, \\ & [\![\sigma_{ij}n_j]\!] = 0 \quad \text{on } \Gamma, \\ & \sigma_{ij}n_j = T_i^0 \quad \text{on } \partial\Omega_N. \end{aligned} \tag{3.1}$$

Introducing a Lagrange multiplier u defined on Ω associated with the first constraint, and another Lagrange multiplier v defined on $\Gamma \cup \partial\Omega_N$ associated with the last two constraints, the above minimization problem is equivalent to the following saddle point problem :

$$\max_{u,v} \min_{\sigma} \mathcal{L}(\sigma, u, v), \tag{3.2}$$

where the system Lagrangian is given by :

$$\begin{aligned} \mathcal{L}(\sigma, u, v) = & \int_\Omega \frac{1}{2}\sigma_{ij}S_{ijkl}\sigma_{kl}d\Omega - \int_{\partial\Omega_D} \sigma_{ij}n_j u_i^0 dS \\ & + \int_\Omega (\sigma_{ij,j} + f_i)u_i d\Omega \\ & + \int_\Gamma [\![\sigma_{ij}n_j]\!]v_i dS + \int_{\partial\Omega_N} (\sigma_{ij}n_j - T_i^0)v_i dS. \end{aligned} \tag{3.3}$$

The first-order optimality conditions of this min/max result in the following mixed varia-

tional formulation : Find $(\sigma, u, v) \in \mathcal{V}_\sigma \times \mathcal{V}_u \times \mathcal{V}_v$ such that :

$$\begin{aligned} &\int_\Omega \widehat{\sigma}_{ij} S_{ijkl} \sigma_{kl} d\Omega + \int_\Omega \sigma_{ij,j} u_i d\Omega \\ &+ \int_\Gamma [\![\widehat{\sigma}_{ij} n_j]\!] v_i dS + \int_{\partial\Omega_N} \widehat{\sigma}_{ij} n_j v_i dS = \int_{\partial\Omega_D} \widehat{\sigma}_{ij} n_j u_i^0 dS \qquad \forall \widehat{\sigma} \in \mathcal{V}_\sigma, \end{aligned} \tag{3.4}$$

$$\int_\Omega \sigma_{ij,j} \widehat{u}_i d\Omega = -\int_\Omega f_i \widehat{u}_i d\Omega \qquad \forall \widehat{u} \in \mathcal{V}_u, \tag{3.5}$$

$$\int_\Gamma [\![\sigma_{ij} n_j]\!] \widehat{v}_i dS + \int_{\partial\Omega_N} \sigma_{ij} n_j \widehat{v}_i dS = \int_{\partial\Omega_N} T_i^0 \widehat{v}_i dS \qquad \forall \widehat{v} \in \mathcal{V}_v, \tag{3.6}$$

where $\mathcal{V}_\sigma$, $\mathcal{V}_u$ and $\mathcal{V}_v$ are appropriate function spaces for the corresponding variable.

Let us introduce the following notations for the different bilinear forms :

$$\begin{aligned} a(\widehat{\sigma}, \sigma) &:= \int_\Omega \widehat{\sigma}_{ij} S_{ijkl} \sigma_{kl} d\Omega, \\ b(\widehat{\sigma}, u) &:= \int_\Omega \widehat{\sigma}_{ij,j} u_i d\Omega, \\ c(\widehat{\sigma}, v) &:= \int_\Gamma [\![\widehat{\sigma}_{ij} n_j]\!] v_i dS + \int_{\partial\Omega_N} \widehat{\sigma}_{ij} n_j v_i dS, \end{aligned}$$

and :

$$\begin{aligned} \ell_1(\widehat{\sigma}) &:= \int_{\partial\Omega_D} \widehat{\sigma}_{ij} n_j u_i^0 dS, \\ \ell_2(\widehat{u}) &:= -\int_\Omega f_i \widehat{u}_i d\Omega, \\ \ell_3(\widehat{v}) &:= \int_{\partial\Omega_N} T_i^0 \widehat{v}_i dS, \end{aligned}$$

for the different linear forms, such that the variational formulation can be rewritten as :

$$\begin{aligned} a(\widehat{\sigma}, \sigma) + b(\widehat{\sigma}, u) + c(\widehat{\sigma}, v) &= \ell_1(\widehat{\sigma}) \qquad \forall \widehat{\sigma} \in \mathcal{V}_\sigma, \\ b(\sigma, \widehat{u}) &= \ell_2(\widehat{u}) \qquad \forall \widehat{u} \in \mathcal{V}_u, \\ c(\sigma, \widehat{v}) &= \ell_3(\widehat{v}) \qquad \forall \widehat{v} \in \mathcal{V}_v. \end{aligned} \tag{3.7}$$

The above symmetric block-like structure is typical of hybrid mixed methods. Let us note that, from a mechanical stand point, u and v can both be interpreted as displacements, defined either in Ω or on $\Gamma \cup \partial\Omega_N$. Finally, no continuity conditions across Γ are required for both σ and u.

3.5.2 Finite-element discretization

Let us now consider a discretization of Ω into a mesh of triangular/tetrahedral cells K and with Γ denoting now the inner facets (segments in 2D, triangles in 3D) of this mesh. It is important to point out that no continuity conditions have to be enforced on the stress variable σ and the Lagrange multiplier u across the mesh cells. They are both defined cell-wise. The Lagrange multiplier v lives on the mesh facets and is not defined inside the cells. No continuity at the mesh vertices (in 2D) or edges (in 3D) linking different facets are required.

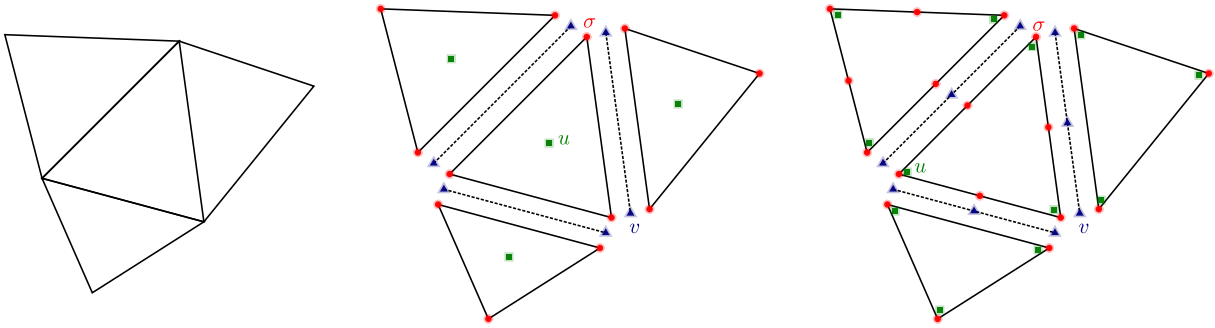


FIGURE 3.1 – Left : geometrical 2D mesh. Middle : mixed approach with $p = 1$. Right : mixed approach with $p = 2$

As a result, let us consider a discretization of the stress field σ using discontinuous Lagrange elements of degree p for $p \geq 1$ (see also Figure 3.1). Enforcing the local balance equation will therefore require an interpolation of the Lagrange multiplier field u using discontinuous Lagrange elements of degree $p-1$. Similarly, stress continuity can be achieved using discontinuous Lagrange elements on the facets of degree p for the multiplier field v.

Using standard finite-element technology, the various bilinear/linear forms involved in (3.7) are assembled into global matrices/vectors forming the following final linear system :

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ \mathbf{B}^T & 0 & 0 \\ \mathbf{C}^T & 0 & 0 \end{bmatrix} \begin{Bmatrix} \Sigma \\ U \\ V \end{Bmatrix} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \tag{3.8}$$

The main feature of hybridizable mixed methods is that operators $\mathbf{A}$ and $\mathbf{B}$ have a block structure since their assembly on each cell involves only stress σ and displacement u variables associated with the cell itself. Both σ and u can therefore be eliminated by local static condensation at the cell level by inverting a small-size matrix. Doing so, the final

system is of much smaller size than (3.8) since it involves only the vector of unknowns V corresponding to the Lagrange multiplier v :

$$\widetilde{\mathbf{A}} V = L, \tag{3.9}$$

where $\widetilde{\mathbf{A}}$ and L are assembled from the corresponding local contribution of cell K :

$$\widetilde{\mathbf{A}}_K = -\begin{bmatrix} \mathbf{C}_K^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}_K & \mathbf{B}_K \\ \mathbf{B}_K^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_K \\ 0 \end{bmatrix} \tag{3.10}$$

$$L_K = (L_3)_K - \begin{bmatrix} \mathbf{C}_K^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}_K & \mathbf{B}_K \\ \mathbf{B}_K^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} (L_1)_K \\ (L_2)_K \end{bmatrix} \tag{3.11}$$

A last interesting feature of hybridized mixed solutions is related to the reconstruction of a displacement field. One simple strategy is to exploit the Lagrange multiplier field u which gives a piecewise polynomial approximation of the real displacement. Projection of u onto a suitable continuous functional space will therefore give a continuous approximation of the displacement. However, as discussed in (Arnold and Brezzi (1985)), the facet Lagrange multiplier v can also be used to derive an even more accurate approximation of the displacement. Such a reconstruction requires the resolution of local problems at the cell-level, the so-obtained displacement being usually non-conforming with continuity at the edge Gauss points. This more advanced reconstruction procedure will not be investigated in this work.

3.6 Hybridization of a mixed method for the SCLS1 model

In this section, we transpose the hybridization of mixed methods as described in the 3D continuum case to the SCLS1 model.

3.6.1 Continuous formulation

Let us first denote by Σ the vector of generalized stresses of the SCLS1 model as described in section 1.3.

The set of equilibrium equations (1.11) will be denoted by $\mathcal{D}\Sigma = f$ where $\mathcal{D}$ is the corresponding linear differential operator and f is related to the imposed values $T_\alpha^\pm, T_3^\pm$ of τ_α and ν on the top and bottom interfaces as in (1.20). Similarly to the 3D continuum, two Lagrange multiplier fields will be introduced. These fields are denoted by U and V and will be respectively used to enforce the generalized equilibrium equations (1.11) and the

continuity and stress boundary conditions associated with (1.12). The latter conditions will be denoted as follows :

$$[\![\mathcal{T}\Sigma]\!] = 0 \quad \text{on } \Gamma, \tag{3.12}$$

$$\mathcal{T}\Sigma = 0 \quad \text{on } \partial\omega_N. \tag{3.13}$$

Finally, introducing $\mathbf{S}$ the generalized compliance matrix involved in the generalized constitutive equations (1.16)-(1.22), the complementarity energy principle characterizing the solution to the elastic SCLS1 model can be stated as :

$$\begin{aligned} \min_{\Sigma} \quad & \int_\omega \frac{1}{2}\Sigma^T \mathbf{S}\Sigma d\omega, \\ \text{s.t.} \quad & \mathcal{D}\Sigma = f \quad \text{in } \omega, \\ & [\![\mathcal{T}\Sigma]\!] = 0 \quad \text{on } \Gamma, \\ & \mathcal{T}\Sigma = 0 \quad \text{on } \partial\Omega_N, \end{aligned} \tag{3.14}$$

in which we considered purely homogeneous Dirichlet and Neumann boundary conditions for simplicity.

The SCLS1 mixed approach will be therefore given by :

$$\begin{aligned} a(\widehat{\Sigma},\Sigma) + b(\widehat{\Sigma},U) + c(\widehat{\Sigma},V) &= 0 \qquad \forall \widehat{\Sigma} \in \mathcal{V}_\Sigma, \\ b(\Sigma,\widehat{U}) &= \ell(\widehat{U}) \qquad \forall \widehat{U} \in \mathcal{V}_U, \\ c(\Sigma,\widehat{V}) &= 0 \qquad \forall \widehat{V} \in \mathcal{V}_V, \end{aligned} \tag{3.15}$$

where :

$$\begin{aligned} a(\widehat{\Sigma},\Sigma) &:= \int_\omega \widehat{\Sigma}^T \mathbf{S}\Sigma d\omega, \\ b(\widehat{\Sigma},U) &:= \int_\omega (\mathcal{D}\widehat{\Sigma})^T U d\omega, \\ c(\widehat{\Sigma},V) &:= \int_\Gamma ([\![\mathcal{T}\widehat{\Sigma}]\!])^T V dS + \int_{\partial\Omega_N} (\mathcal{T}\widehat{\Sigma})^T V dS, \\ \ell(\widehat{U}) &:= \int_\omega f^T \widehat{U} d\omega. \end{aligned}$$

3.6.2 Finite-element implementation

Throughout this study, we will consider the following discretization strategy for the hybridizable mixed approach on a triangular mesh :

— discontinuous Lagrange interpolation of degree p for Σ

— discontinuous Lagrange interpolation of degree $p-1$ for U

— discontinuous Lagrange interpolation of degree p on edges for V
with either $p = 1$ or $p = 2$ in the subsequent numerical examples.

This choice is similar to the 3D continuum case although it is not obvious that this choice will be numerically stable. Let us mention that we will not necessarily expect similar stability results as in the 3D continuum case since the generalized differential operator $\mathcal{D}$ couples the various generalized stress fields of Σ through first-order and zero-order derivatives as seen in equation (1.11). Numerical analysis of the chosen discretization for the SCLS1 model is out of the scope of the present work and stability will only be assessed numerically in the next section.

However, the discontinuous nature of the chosen interpolations will make it possible to reduce the system at the cell level through local static condensation as discussed in the 3D case. The final reduced system also involves only the Lagrange multiplier field V. Table 3.1 enumerates the total number of degrees of freedom per triangular cell with and without static condensation. We also compare the resulting size with a displacement-based interpolation using continuous quadratic Lagrange triangles for the generalized displacement field as in (Baroud et al. (2016); Salha et al. (2020)). Clearly, static condensation is absolutely necessary to obtain a reasonable dof count for the mixed approach. For example, for $p = 2$ and for 5 layers, without static condensation there would be around 600 dofs/triangle against 135 with static condensation. It can also be observed that both mixed approaches with static condensation are more expensive than the quadratic displacement approach of (Baroud et al. (2016); Salha et al. (2020)).

Let us recall that solving the full system of the mixed approach or the reduced system with static condensation yields exactly the same results. The only difference lies in the numerical resolution which is more efficient in the reduced system with static condensation since the final system involves less degrees of freedom. One can however always back-substitute the reduced solution to recover the initial degrees of freedom of the mixed approach. In the subsequent part of this manuscript, when referring to results of the mixed approach, computations will always be performed by solving the reduced system after condensation.

Discretization	Σ	U	V	Total	Condensed
Mixed $(p=1)$	$3(12n-4)$	$6n-1$	$2\frac{3}{2}(6n-1)$	$\approx 60n$	$\approx 18n$
Mixed $(p=2)$	$6(12n-4)$	$3(6n-1)$	$3\frac{3}{2}(6n-1)$	$\approx 117n$	$\approx 27n$
Displacement	–	–	–	$2(6n-1) \approx 12n$	

TABLE 3.1 – Number of degrees of freedom per cell for both mixed discretizations and a pure displacement approach. Each vertex dof counts for $\frac{1}{6}$ in a cell and each edge dof for $\frac{1}{2}$ in the asymptotic fine mesh regime. Total and condensed numbers of dofs are approximated for large values of n.

Finally, we compare the number of degrees of freedom between two finite elements discretization : the quadratic mixed discretization applied to the LS1 model (M4-5n approach) and the finite element discretization considered by (Nasser et al. (2018a)) for the M4-5n model as shown in Figure 3.2.

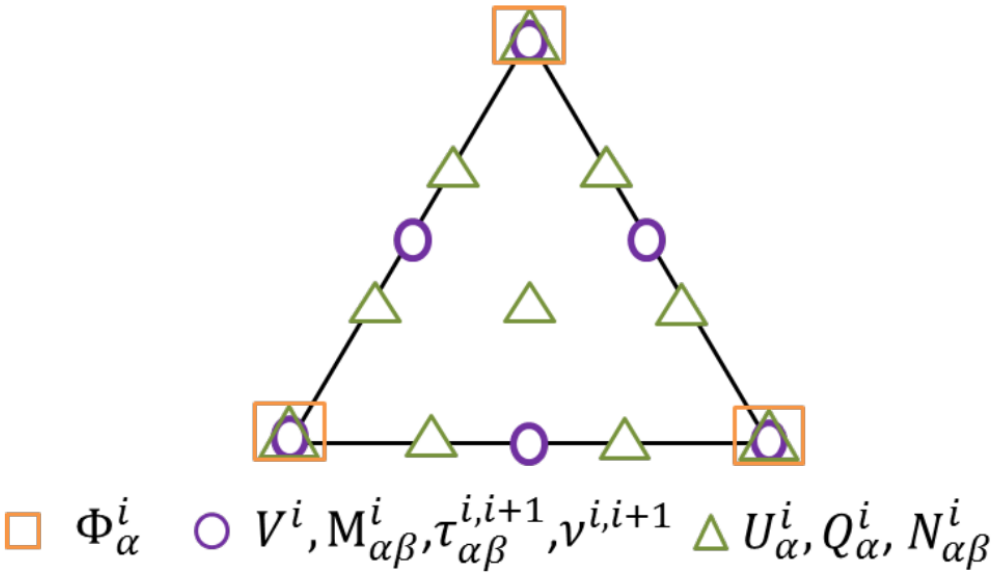


FIGURE 3.2 – M4-5n mixed finite element with multiple nodes in Nasser et al. (2018a)

Discretization	Σ	U	V	Total	Condensed
LS1 (Mixed $p=2$)	$6(11n-3)$	$3(5n)$	$2\frac{3}{2}(5n)$	$\approx 103n$	$\approx 22n$
Nasser(Figure 3.2)	$86n-16$	$32n$	–	$\approx 118n$	–
Nasser(P1-bubble/P1)	–	–	–	$\approx 59n$	$\approx 48n$

TABLE 3.2 – Number of degrees of freedom per cell for mixed discretization and both discretizations used by (Nasser et al. (2018a)).

The previous Table 3.2 shows the difference in the number of degrees of freedom between our quadratic mixed discretization and the one used by Nasser. Our discretization allows us to condense the dof, and by comparison we found that the number of dof per cell in our study is much lower than the one obtained by [Nasser et al. (2018a)].

Numerical implementation has been performed using the Firedrake software package (Rathgeber et al. (2016)). The local static condensation operations are performed automatically using the Slate domain-specific language (Gibson et al. (2020)). We used a recent version of Firedrake which relies on Loopy (Klöckner (2014)) for an optimized assembly of matrix-free local finite-element kernels (Kirby and Mitchell (2018); Sun et al. (2019)). The main advantage of using finite-element softwares like FEniCS, Firedrake or FreeFem++ is that the user only has to formulate the corresponding variational formulation and select the appropriate discretization spaces. There is no need to write elementary stiffness matrices for instance, as such computations are taken care of by the software using C functions which are automatically generated and compiled on the fly.

3.7 Illustrative applications

In this section, we investigate different illustrative applications assessing the quality of the hybridized mixed approach.

3.7.1 Homogeneous laminate

This first example considers a homogeneous thin (resp. thick) square plate of length $l = 1$ and thickness $h = 0.01$ (resp. $h = 0.2$). The constitutive material is assumed to be isotropic with $E = 10$ GPa and $\nu = 0.3$. The plate is fully clamped on its boundary and subjected to a uniform vertical loading of density $q = 8$. Calculations are performed considering a subdivision of the plate thickness in $n = 2$ layers. To compare the performance of the mixed approach with the more standard displacement-based FE interpolation, we monitor the evolution of the total elastic energy ($\frac{1}{2}a(\Sigma, \Sigma)$ for the present mixed approach) with mesh refinement.

In the thick plate case ($h = 0.2$), convergence of the total energy has been represented in Figure 3.3 in terms of total number of degrees of freedom. One can observe that convergence of the quadratic mixed approach ($p = 2$) is faster than the linear mixed approach ($p = 1$) and the displacement approach. In particular, the total energy is already well approximated with the coarsest mesh (2 elements/side) unlike the displacement approach.

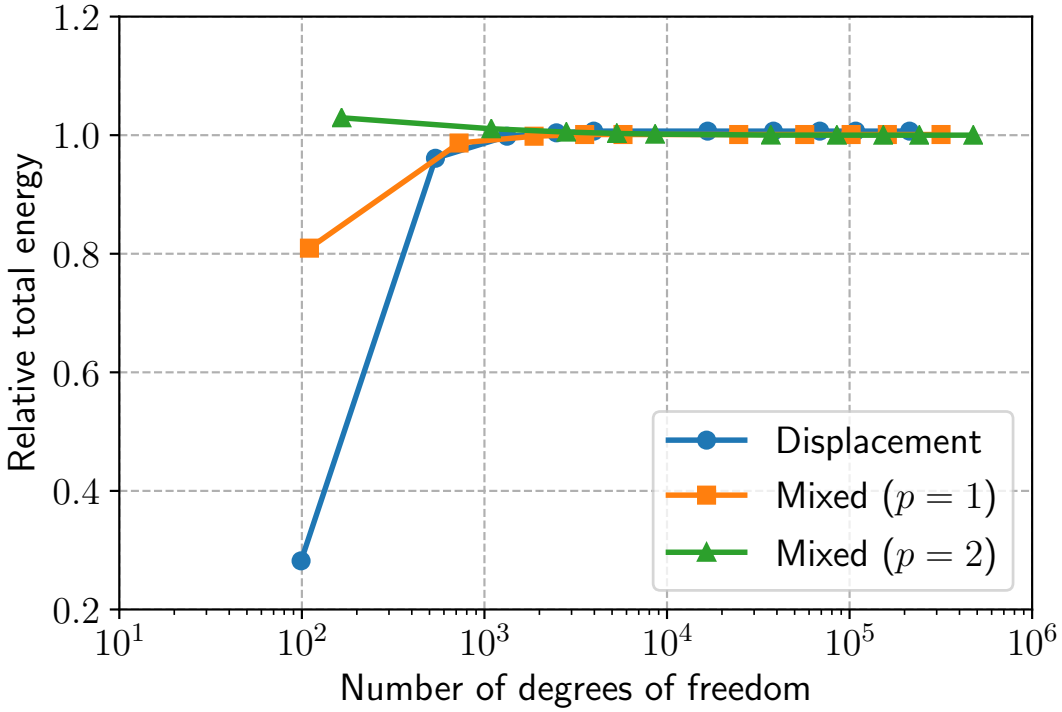


FIGURE 3.3 – Total energy convergence for the clamped thick plate case

The corresponding results in the thin plate case ($h = 0.01$) have been presented in Figure 3.4. In this case, the quadratic mixed approach still exhibits a faster convergence than the displacement approach. However, in this case, the linear mixed approach shows an extremely slow convergence rate. This may be attributed to a lack of stability of the retained discretization choice which would require more in-depth mathematical analysis. As a result, only the mixed approach with $p = 2$ will be retained in the remaining part of the chapter. Finally, we can remark that, unlike the displacement approach, the quadratic mixed approach exhibits no shear-locking effect in the thin plate limit. Specific treatment such as selective reduced integration as in the displacement approach is therefore unnecessary.

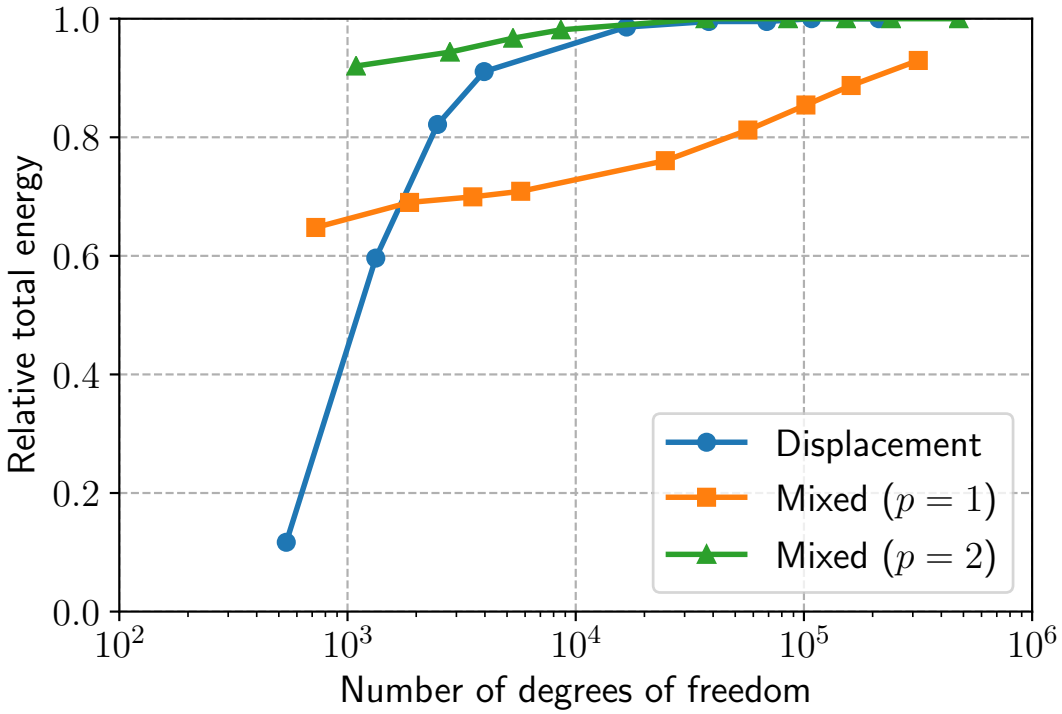


FIGURE 3.4 – Total energy convergence for the clamped thin plate case

Figures 3.5 and 3.6 represent respectively the results of the linear mixed approach ($p = 1$) and the quadratic mixed approach ($p = 2$) for three different thicknesses ($h = 0.001, h = 0.01), h = 0.2$). The case of linear mixed approach shows clearly the shear-locking for the thin plate, while the quadratic mixed approach demonstrates and shows the no shear-locking effect in the thin plate. Finally, we can also observe that the quadratic mixed approach, although exhibiting good convergence irrespective of the plate thickness, tends to converge from above for thick plates and from below for thinner plates. This is a confirmation that our approach is of mixed nature : it is neither a pure displacement (which would always converge from above) nor a pure static approach (which would always converge from below). We can postulate that in the thin plate case, shear effects become negligible and our mixed approach almost satisfies exactly the equilibrium and traction continuity conditions related to the bending part which results in a static-like convergence behaviour.

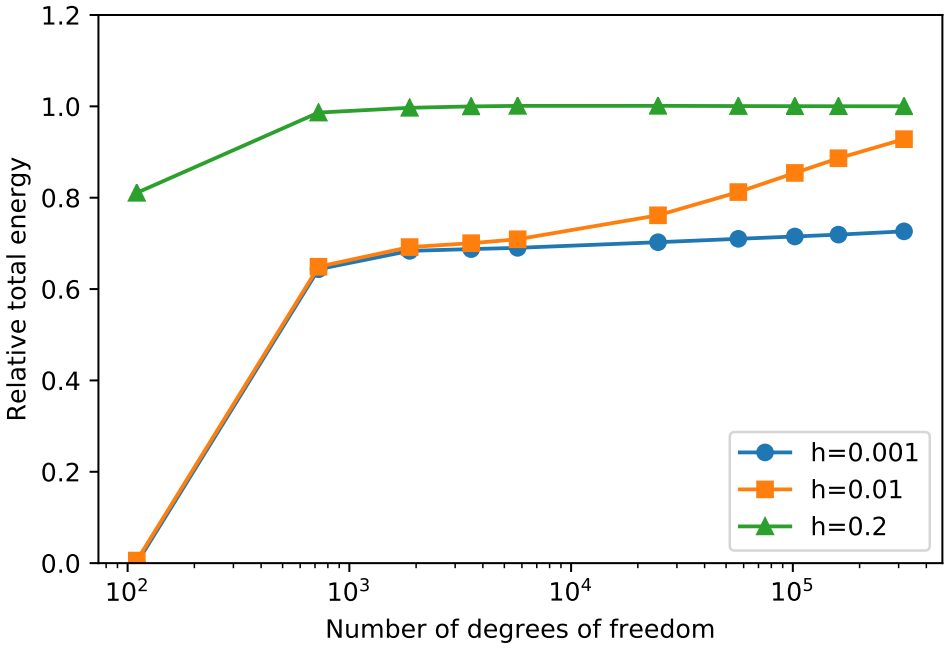


FIGURE 3.5 – Total energy for quadratic mixed approach (p=1) for different thicknesses.

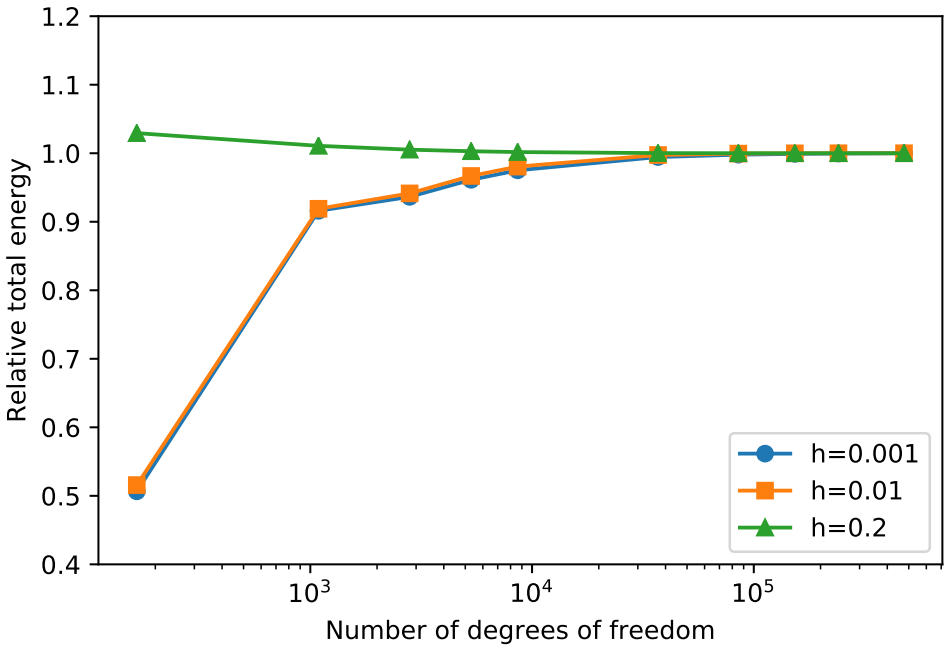


FIGURE 3.6 – Total energy for quadratic mixed approach (p=2) for different thicknesses.

Let us now compare the values of the vertical deflection U_3 along the plate middle line $0 \leq x \leq 1, y = 0.5$ for the thin plate $h = 0.01$. In the displacement approach, U_3 is one of the primal unknowns whereas in the quadratic mixed approach the deflection is computed from an L^2-projection of the corresponding cell Lagrange multiplier U on the space of piecewise linear Lagrange polynomials. We represent on Figure 3.7 the deflection of the first layer (in this case, both layers have the same deflection) on a coarse (10 elements/side) and fine (70 elements/side) mesh. It can clearly be observed that, for a coarse mesh, the projected deflection obtained from the mixed approach is more accurate than the deflection computed from the displacement approach. For a fine mesh, both solutions coincide as expected.

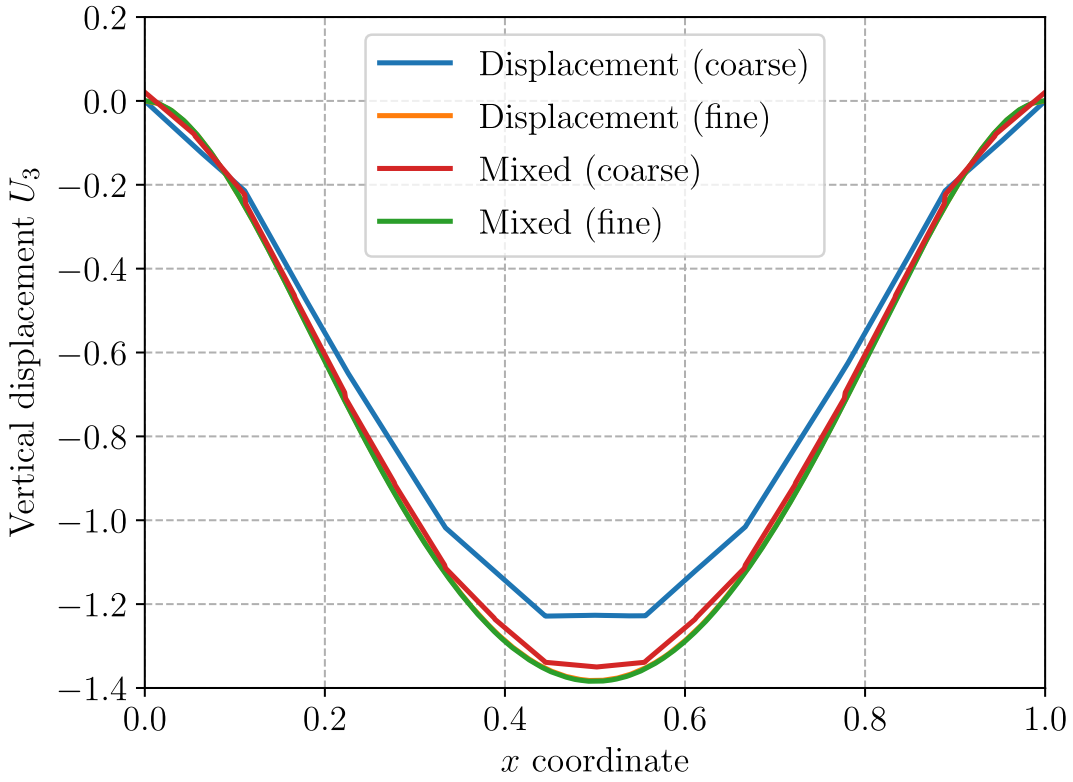


FIGURE 3.7 – Vertical deflection U_3 for the thin plate case (both solutions in the fine mesh case are superimposed)

Finally, a similar comparison is performed regarding the horizontal membrane force N_{11} in the first layer along the same middle line. As regards the displacement approach, the membrane force computed from the FE displacement solution will, a priori, be discontinuous across cells. We therefore represent, for the coarse mesh, a discontinuous version N_{11} projected over a piecewise constant space as well as a "smoothed" version obtained by projection N_{11} over a continuous piecewise linear space. Such stress fields obtained from a displacement FE approach are not very accurate for a coarse mesh contrary to their

counterpart obtained with a mixed approach. The projecting over a continuous space even deteriorates the quality of the approximation of the discontinuous version. This is a clear advantage of mixed methods since stress fields are usually the quantities of interest used by the engineer to design mechanical systems. Obtaining more accurate estimations of stresses on a coarse mesh is thus extremely valuable.

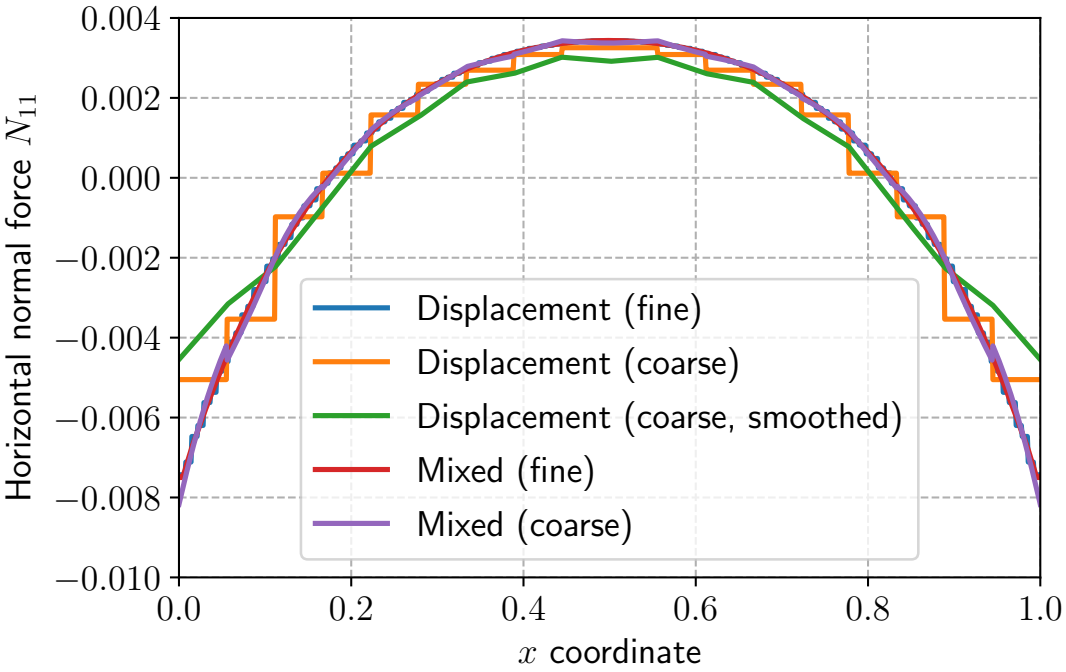


FIGURE 3.8 – Horizontal membrane force N_{11} for the thin plate case

3.7.2 Laminate with a circular hole

The second example considers a rectangular multilayered plate of length $L = 6$, width $W = 1$ and total thickness $h = 0.2$. The plate is perforated by a circular hole of radius $R = 0.15$ in its center. The laminate is made of a transversely isotropic material of elastic properties $E_T = 14.48GPa$, $E_L = 137.9GPa$, $\nu_T = 0.21$, $\nu_L = 0.21$, $\mu_T = 5.86GPa$ and $\mu_L = 5.86GPa$ with L (resp. T) denoting the fiber longitudinal direction (resp. the perpendicular transverse direction). The laminate consists of 5 plies (one layer per ply) with fiber oriented at $[90°, 45°, 0°, -45°, 90°]$ with respect to the horizontal direction. The plate is clamped on the left boundary, free on the top, bottom and on the hole boundaries. A uniform traction is applied to the 3rd layer on the right boundary while the other layers remain free as shown in Figure 3.9. Due to the laminate anisotropic layout, strong stress concentrations are expected at the hole boundary. Similarly, applying a traction only on the third layer will induce a boundary layer effect along which stress transfers will occur between the third layer and its adjacent ones.

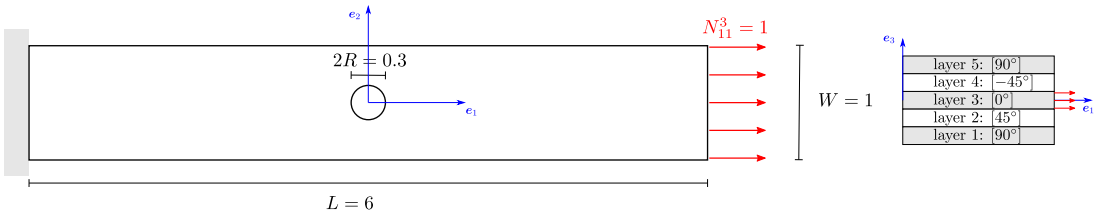


FIGURE 3.9 – Laminate with a circular hole under tension

As expected, Figure 3.10 shows that each individual layer experiences strong stress concentrations near the hole boundary. The resulting stress fields clearly illustrate the material anisotropy between the first, second and third layer.

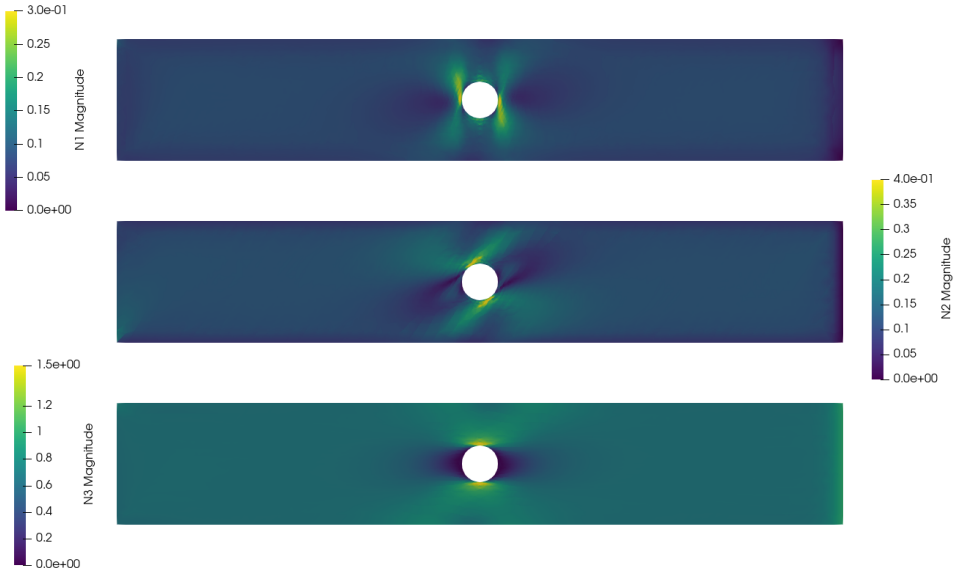
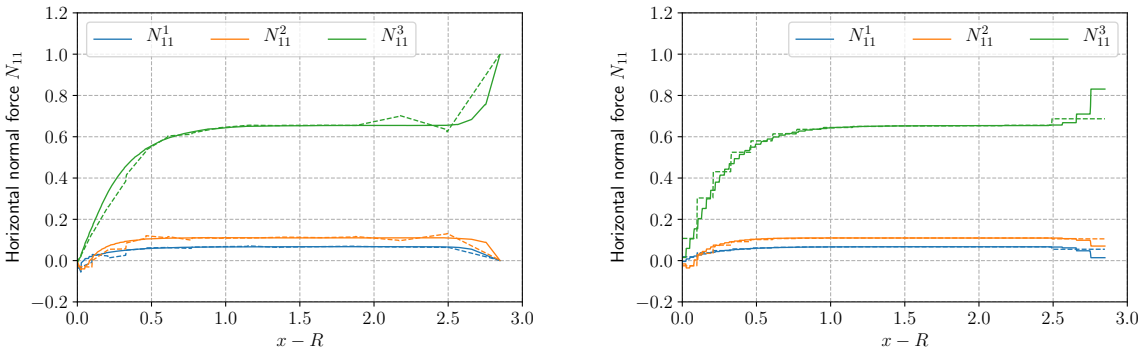


FIGURE 3.10 – Magnitude of $N_{\alpha\beta}$ in layers 1, 2, and 3

In addition, we compare the N_{11} membrane force in the first three layers along the $y = 0$, $x \geq R$ line on Figure 3.11 for both mixed and displacement approaches for a coarse and a fine mesh. Clearly, the mixed approach succeeds in satisfying the stress-free boundary condition on the right boundary in layers 1 and 2, even with a coarse mesh (dotted line). The edge effects near the boundary are therefore much better represented than using a displacement approach, even with a fine mesh. This feature is particularly beneficial in order to accurately predict the occurrence of delamination in composite laminates for instance.



(a) N_{11} with the mixed ($p = 2$) approach (b) N_{11} with the displacement approach

FIGURE 3.11 – Evolution of N_{11} along the plate length for a coarse (dotted line) and a fine (solid line) mesh

3.7.3 Bending of a laminate with multi-cracking

The final example (Figure 3.12) considers a rectangular multilayered plate having the same dimensions as previously 3.7.2(without the circular hole). The laminate consists of 5 plies (one layer per ply) with fiber oriented at $[0°, 90°, 0°, 90°, 0°]$ with respect to the horizontal direction and with the same lamination properties. The second layer is weakened by the introduction of 5 cracks through the ply thickness. The plate is clamped at the left border, free on its lateral sides and a uniform vertical force is applied on the right border for all layers i.e. $Q_1^i = 1$ for $i = 1, \ldots, 5$.

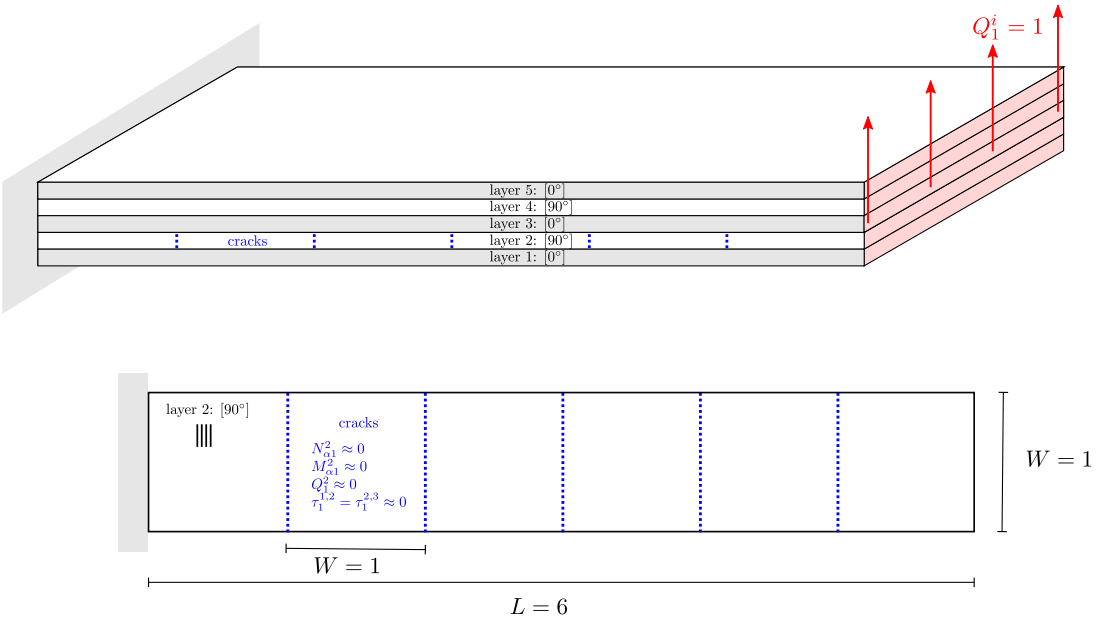


FIGURE 3.12 – cracked plate problem

Cracks are modeled by adding to the variational formulation (3.15) an additional elastic energy of the corresponding generalized tractions on the crack interface Γ_{crack}, i.e. the bilinear form a representing the elastic energy is replaced by :

$$\tilde{a}(\widehat{\Sigma}, \Sigma) = a(\widehat{\Sigma}, \Sigma) + a_{\text{crack}}(\widehat{\Sigma}, \Sigma), \tag{3.16}$$

$$a_{\text{crack}}(\widehat{\Sigma}, \Sigma) = \frac{1}{K_{\text{int}}} \int_{\Gamma_{\text{crack}}} \left(\widehat{N}^2_{\alpha 1} N^2_{\alpha 1} + \widehat{M}^2_{\alpha 1} M^2_{\alpha 1} + \widehat{Q}^2_1 Q^2_1 + \widehat{\tau}^{1,2}_1 \tau^{1,2}_1 + \widehat{\tau}^{2,3}_1 \tau^{2,3}_1 \right) dS, \tag{3.17}$$

where the associated interface stiffness is assumed to be very small $K_{int} \ll 1$ (note that we did not pay attention to physical units in this penalized term).

Figure 3.13 represents the variations of the generalized stresses N^2_{11}, Q^2_1, $\tau^{2,3}_1$ and the vertical displacement U^2_3 in layer 2. As expected, the stress fields exhibit strong variations around the cracks. Indeed, these fields must vanish at the crack stress-free interface, but due to the bonding exerted by adjacent layers, stress transfers occur between the different cracked regions. The stress fields recover a value similar to what it would be without the cracks. On the contrary, the vertical displacement field remains continuous whereas the in-plane displacement U^2_1 in layer 2 exhibits weak discontinuities due to the bonding of adjacent layers. As a result, a mixed approach is extremely advantageous in such situations where stresses are much more singular than displacements.

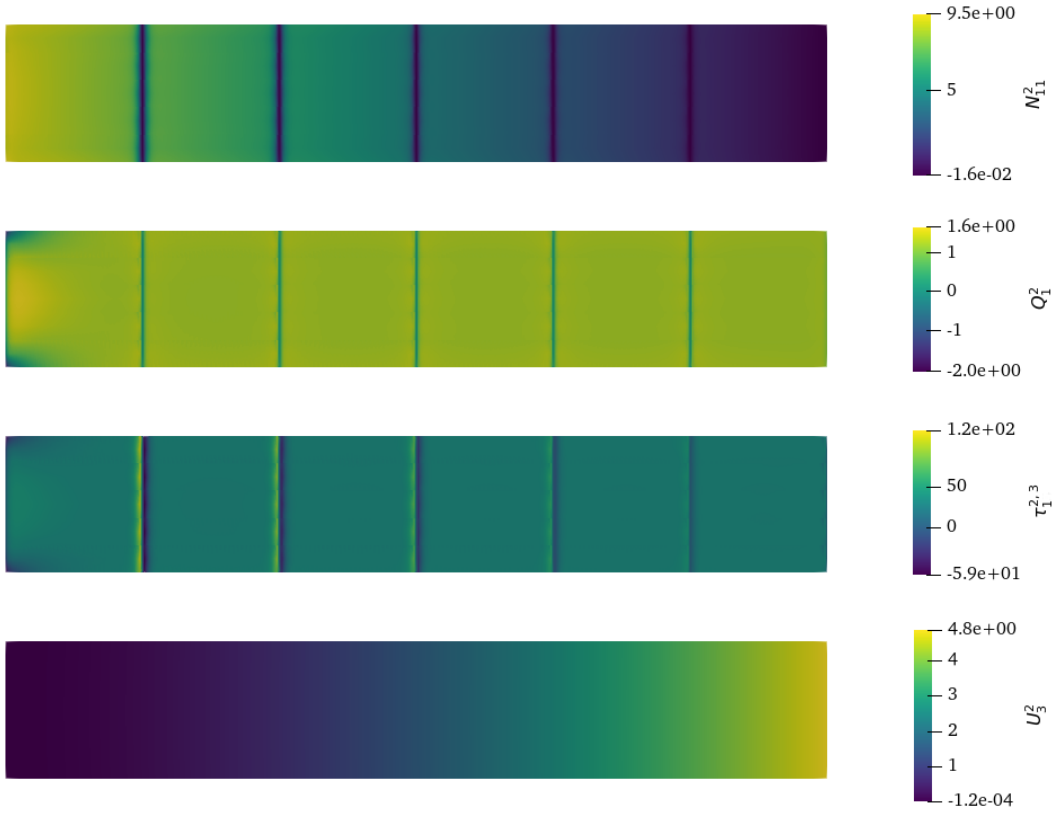
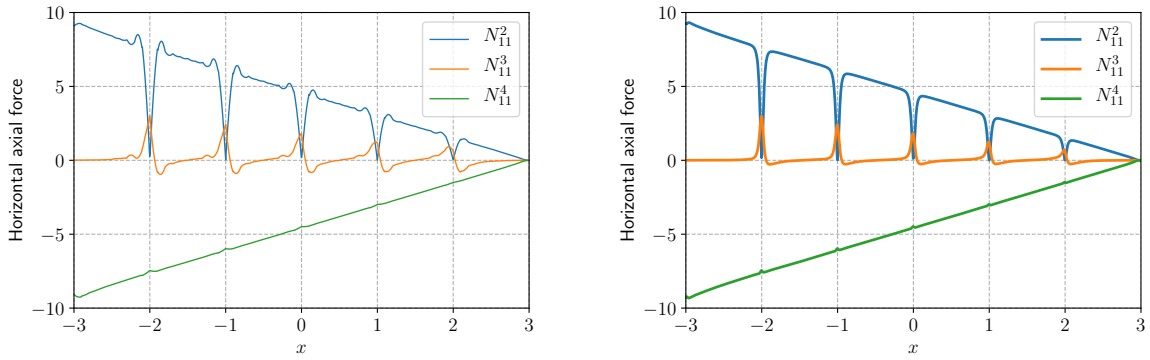


FIGURE 3.13 – Generalized stresses and vertical displacement fields in layer 2

The variation of the axial force N_{11} in layers 2, 3 and 4 for a coarse mesh (5 elements between cracks) is represented in Figure 3.14a. When compared to the solution on a much finer mesh (Figure 3.14b), we notice that the strong variations near the cracks are already well captured.

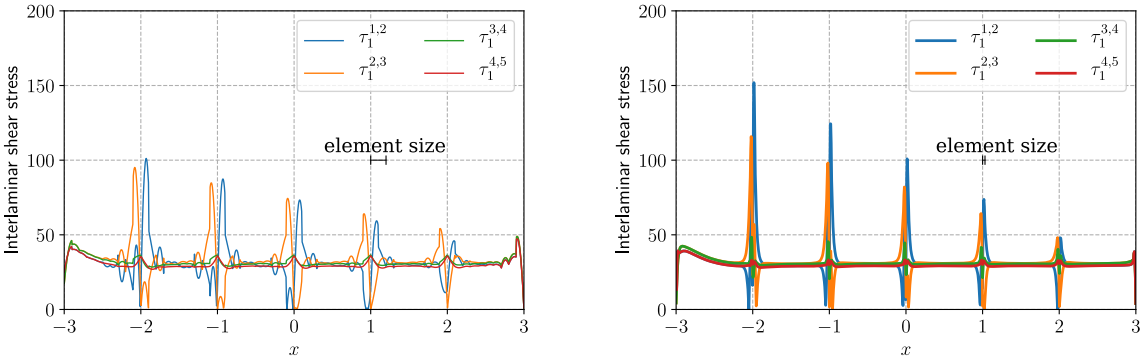


(a) Coarse mesh

(b) Fine mesh

FIGURE 3.14 – Axial force N_{11} along the plate length in layers 2, 3 and 4

Finally, the variations of the interfacial shear stress field τ_1 at the interfaces $(1,2)$, $(2,3)$, $(3,4)$ and $(4,5)$ are represented in Figure 3.15. Such strong variations are harder to capture by the coarse mesh, especially when looking at the maximum values which are the quantities of interest that will drive the occurrence of interface delamination in such situations. However, we observe that coarse meshes can be used with the mixed approach to obtain an accurate estimation of such interfacial stresses.



(a) Coarse mesh

(b) Fine mesh

FIGURE 3.15 – Interfacial stress τ_1 along the plate length at all interfaces

3.8 Conclusion and perspectives

In this chapter, a mixed finite-element discretization has been considered as an alternative to the conventional displacement-based finite-element method. A hybridization strategy has been considered in which traction continuity is enforced explicitly through the use of an additional displacement-like Lagrange multiplier defined on the elements facets. Such a strategy offers the advantages of uncoupling many degrees of freedom. Local static condensation can be performed to yield a much smaller final system to solve. This static condensation operation on such a complex mechanical model has been made possible by recent developments in automated code-generating finite-element solvers such as Firedrake. Finally, some illustrative applications demonstrate that the proposed mixed approach is free from any shear-locking in the thin plate limit and is more accurate than a displacement approach for the same number of freedom degrees. This mixed-method is therefore particularly well suited to capture strong intra-laminar and inter-laminar stress variations near free edges and cracks. Further developments might be interesting to pursue. For instance, it is clear that such a complex layerwise model is efficient and relevant in regions of strong stress variations (near cracks, holes, loading application, etc.). However, in the bulk region, i.e. far from these regions, it is known that an equivalent single-layer model might be enough to describe the behavior of the laminate. An interesting perspective would then to couple a detailed layerwise model in the previously mentioned critical regions with an ESL model in the bulk region. In this respect, the approach of (Vidal

et al. (2013)) using the Proper Generalized Decomposition seems promising.

Finally, another natural extension of the present work is to model delamination phenomena between the different layers and simulate their propagation. A specific attention should therefore be paid to how delamination might impact the efficiency of the static condensation procedure.

Conclusions and perspectives

The good implementation of multi-layered structures involves numerical tools that comply with their specific heterogeneity. The multi-layered plate is conveniently represented as a uniform, anisotropic bundle by the structural engineer. The free-edges effects are one of the key problems with the design and study of this plate. The variations in elastic characteristics of the adjoining layers have been demonstrated to produce high concentrations of interlaminar stresses close to open borders. This can lead to delaminations that can cause the multilayered structure to fail globally. Given the industry's strong interest in reliable models, several suggestions were made. The main aim is to simplify the computationally heavy 3D model into a 2D plate model without losing the precision of local 3D fields. First of all, it should be easy to use plate theory with a regular finite element program and a good relocation of the 3D fields to measure local stresses, unlimited from the local material symmetries to a good estimating of the macroscopic strain.

In this dissertation, first, we recalled the Statically compatible Layerwise model with first-order membrane stress approximations per layer in thickness direction (SCLS1). The SCLS1 model is proposed in order to enhance the LS1 model by eliminating its disadvantages. The plate is still used in the SCLS1 model as an overlay of Reissner plates with interfacial stresses. The variations between the transverse interlaminar shears are therefore introduced as new general stresses. Then, we developed a mesh adaptation strategy which relies on the reconstruction of 3D displacement fields from the model generalized displacements. In addition, some examples show that this adaptive method paves the way to further developments including interface delamination propagation or damage at the ply level. A second line of work is related with this fact. Although the layerwise model is built at the continuous level form a stress-based perspective complying with the equilibrium equations, its numerical resolution is performed through a displacement-based approximation for the in-plane variations. As a consequence, the resulting generalized stress fields, and therefore, the associated 3D stress field, do not satisfy strongly the equilibrium equations. In order to maintain the initial philosophy of a stress-based statically compatible construction, an alternative finite element discritization to the conventional displacement-based finite-element method was examined. Then, a hybridization strategy in which traction continuity is modified by introducing an additional displacement such as Lagrange multiplier defined on the elements facets was considered. Finally, some examples show that the proposed mixed approach is more exact than the displacement approach, and can be used to capture the strong intra-laminar and inter-laminar stress values va-

riations near free edges and cracks.
An interesting perspective would then couple a detailed layerwise model in the previously mentioned critical regions with an ESL model in the bulk region. Finally, another natural extension of the present work is to model delamination phenomena between the different layers and simulate its propagation. So we should give specific attention to how delamination might impact the efficiency of the static condensation procedure.

Annexes

1 Annex A

The FEniCS Project is a research and software project aimed at creating mathematical methods and software for automated computational mathematical modeling. This means creating easy, intuitive, efficient, and flexible software for solving partial differential equations (PDEs) using finite element methods. FEniCS was initially created in 2003 and is developed in collaboration between researchers from a number of universities and research institutes around the world. For more information about FEniCS and the latest updates of the FEniCS software visit the FEniCS web page at https ://fenicsproject.org.

FEniCS consists of a number of building blocks (software components) that together form the FEniCS software : DOLFIN [101], FFC [88], FIAT [87], UFL [3], mshr, and a few others as shown in Figure 1.

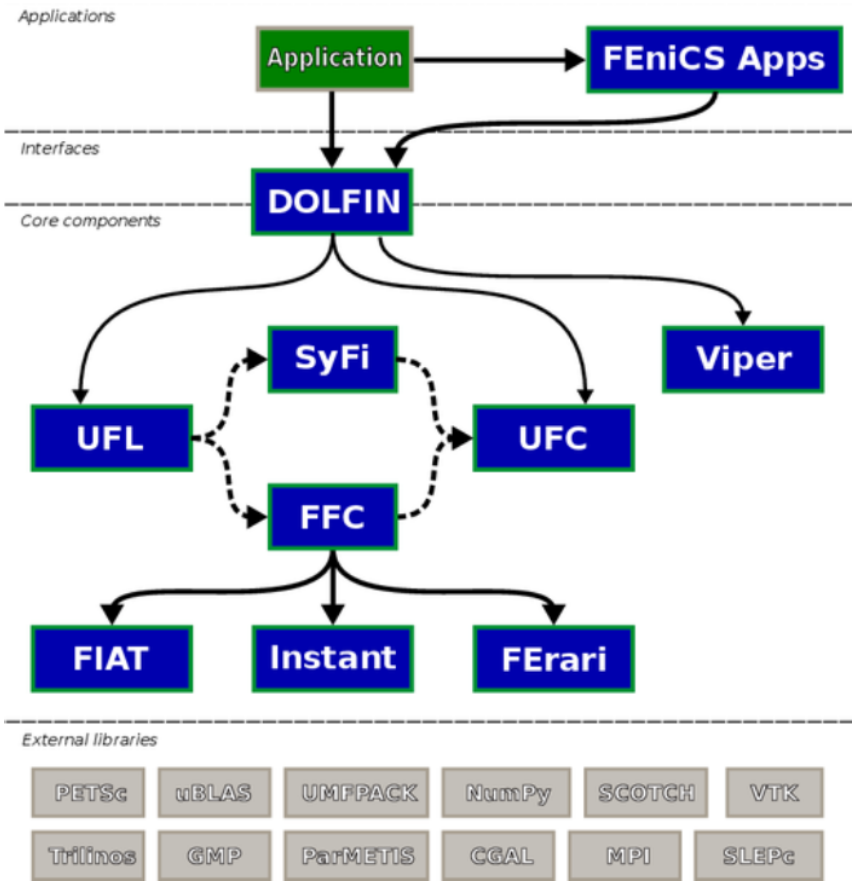


FIGURE 1 – A schematic overview of the FEniCS components and their interplay

Where, the core components are :

- UFL (Unified Form Language), a domain-specific language embedded in Python for specifying finite element discretizations of differential equations in terms of finite

element variational forms. UFL implements the abstract mathematical laguage by which users may express variational problems.

- FIAT (Finite element Automatic Tabulator), the finite element backend of FEniCS, a Python module for generation of arbitrary order finite element basis functions on simplices.
- FFC (FEniCS Form Compiler), a compiler for finite element variational forms taking UFL code as input and generating UFC output. FFC is responsible for generating efficient C++ code from high-level mathematical abstractions.
- UFC (Unified Form-assembly Code), a C++ interface consisting of low-level functions for evaluating and assembling finite element variational forms.
- Instant, a Python module for inlining C and C++ code in Python.
- DOLFIN, a C++/Python library providing data structures and algorithms for finite element meshes, automated finite element assembly, and numerical linear algebra. DOLFIN, the computational high-performance C++ backend of FEniCS. DOLFIN implements data structures such as meshes, function spaces and functions, compute-intensive algorithms such as finite element assembly and mesh refinement, and interfaces to linear algebra solvers and data structures such as PETSc. DOLFIN also implements the FEniCS problem-solving environment in both C++ and Python. Its functionality integrates the other FEniCS components and handles communication with external libraries such as PETSc, Trilinos and Eigen for numerical linear algebra, ParMETIS and SCOTCH for mesh partitioning, and MPI and OpenMP for distributed computing.

For an overview, see [102]. FEniCS users rarely need to think about this internal organization of FEniCS, but since even casual users may sometimes encounter the names of various FEniCS components, we briefly list the components and their main roles in FEniCS.
FEniCS is a user-friendly tool for solving partial differential equations(PDEs). The mathematics of the illustrations is kept simple to better focus on FEniCS functionality and syntax. This means that we mostly use the Poisson equation and the time-dependent diffusion equation as model problems, often with input data adjusted such that we get a very simple solution that can be exactly reproduced by any standard finite element method over a uniform, structured mesh. This latter property greatly simplifies the verification of the implementations. Occasionally we insert a physically more relevant example to remind the reader that the step from solving a simple model problem to a challenge real-world problem is often quite short and easy with FEniCS.

Using FEniCS to solve PDEs may seem to require a thorough understanding of the abstract mathematical framework of the finite element method as well as expertise in Python programming. Nevertheless, it turns out that many users are able to pick up the fundamentals of finite elements and Python programming. You will be amazed at how easy it is to solve PDEs with FEniCS!

2 Annex B

In this annex we presents our code that we used to solve the generalized system of SCLS1 model. This code studies the SCLS1 model and applies our Re-mesh strategy to the 4 illustrative examples shown in 2.5.

```
#Variables

TOL = 5e-4 #Tolerance error
REFINE_RATIO = 0.50 #To Refine 50% of the cells in each iteration
MAX_ITER = 7 #Maximal number of iterations (Homogenoues, Triple laminate
and Double-Cantilever Beam with delaminated interface)
# MAX_ITER = 4 (Plate with circular hole).
iteration = 0
```

In all the examples we begin with coarse mesh. When we use "crossed" that means that the mesh is a structured mesh.

```
###########################################################################
# The mesh structure for Homogeneous and Triple laminate
###########################################################################
N = 2
new_mesh = UnitSquareMesh(N, N, "crossed")# do not lock for continuous

###########################################################################
# The mesh structure for laminate with a circular hole
###########################################################################
L, W = 6, 1
R = 0.15
N1 = 10

domain =Rectangle(Point(-L/2, -W/2), Point(L/2, W/2))-Circle(Point(0., 0.), R)
new_mesh = generate_mesh(domain, N1)

###########################################################################
# The mesh structure for Double-Cantilever Beam with delaminated interface
###########################################################################
n = 6
L, W = 6, 1
new_mesh =  RectangleMesh(Point(0.,0.), Point(L, W), 12, 2, "crossed")

###########################################################################
```

```
# The number of layers
##############################################################################
n = 3 or n = 5 for Homogeneoues and Triple laminate
n = 6 for Laminate with circular hole and DCB with delaminated interface

##############################################################################
# The total thickness of plate is h
##############################################################################
 In the homogeneous laminate :
h = 0.2

 In the Triple laminate :
h2 = 0.12
h1 = 0.04
h = h2 + 2*h1

 In the laminate with circular hole :
h = 0.01

 In the DCB with delaminated interface :
h = 0.01

##############################################################################
# The thickness discretization
##############################################################################
For Triple laminate :
e = [h1]
for i in range(n-2):
    e.append(h2/(n-2))
e.append(h1)

For other examples
e = []
for i in range(n):
    e.append(h/n)

##############################################################################
# The values of h_i^- and  h_i^+   through thickness
(We start by -h/2 to center the plate at 0)
##############################################################################

hb = [-h/2]
for i in range(n):
```

```
    hb.append(hb[i] + e[i])

##############################################################################
# The values of  $\overline{h}_i$
##############################################################################

mh = []
for i in range(n):
    mh.append((hb[i] + hb[i+1])/2)


##############################################################################
# The constitutive materials values and the transveral load
##############################################################################

f = 1e3*h**3
load = np.array([0, 0, 0, 0, -f/2, -f/2])
```

The Constitutive materials values for Homogeneous laminate

```
E = [10.]*n
nu = [0.3]*n
```

The Constitutive materials values for triple laminate

```
E = [50.]
for i in range(n-2):
    E.append(10.)
E.append(50.)

nu = [0.2]
for i in range(n-2):
    nu.append(0.3)
nu.append(0.2)
```

The Constitutive materials values and the orientation angles
for laminate with circular hole and DCB with laminated interface

```
angle =[0*pi/180., 90*pi/180., 45*pi/180., -45*pi/180., 90*pi/180., 0*pi/180.]
EL = 137.9e3
ET = 14.48e3
nuL = 0.21
nuT = 0.21
muL = 5.86e3
```

```
muT = 5.86e3

mu = []
for i in range(n):
    mu.append(E[i]/2/(1+nu[i]))

##########################################################################
The functions that give the indices of the generalized displacement in
the total vector of the generalized displacements [δ].
Where, [δ] = (U_1^1, U_2^1, U_3^1, Φ_1^1, Φ_2^1, ⋯, U_1^n, U_2^n, U_3^n, Φ_1^n, Φ_2^n, V^{1,2}, ⋯, V^{n-1, n}).
##########################################################################

def iUa(n):
    indice_U_alpha = []
    for i in range(n):
        indice_U_alpha.append(3*i)
    return indice_U_alpha

def iU3(n):
    indice_U_3 = []
    for i in range(n):
        indice_U_3.append(3*i+1)
    return indice_U_3

def iP(n):
    indice_Phi_alpha = []
    for i in range(n):
        indice_Phi_alpha.append(3*i+2)
    return indice_Phi_alpha

indice_Vi = []
def iV(n):
    for j in range(n-1):
        indice_Vi.append(j+3*n)
    return indice_Vi

##########################################################################
# The flexibility matrix
##########################################################################
For Homogeneous and Triple laminate
lS_klmn =[]
lS_abgd = []
lS_ab33 = []
```

```
lS_a3b3 = []
lS3333 = []
for i in range(n):
    lS_abgd.append(np.array([[1/E[i], -nu[i]/E[i], 0], [-nu[i]/E[i], 1/E[i], 0],
                    [0, 0, 1/mu[i]]]))
    lS_ab33.append(np.array([[-nu[i]/E[i]], [-nu[i]/E[i]], [0]]))
    lS_a3b3.append(np.array([[1/(4*mu[i]), 0], [0, 1/(4*mu[i])]]))
    lS3333.append(1/E[i])
    lS_klmn.append(np.array([[1/E[i], -nu[i]/E[i], -nu[i]/E[i], 0, 0, 0],
                     [-nu[i]/E[i], 1/E[i], -nu[i]/E[i], 0, 0, 0],
                     [-nu[i]/E[i], -nu[i]/E[i], 1/E[i], 0, 0, 0],
                     [0, 0, 0, 1/mu[i], 0, 0], [0, 0, 0, 0, 1/mu[i], 0],
                     [0, 0, 0, 0, 0, 1/mu[i]]]))
```

For Laminate with circular hole and DCB with delaminated interfaces

```
def rotate(mat, alpha):
new_mat = copy.deepcopy(mat)
new_mat.rotate(alpha)
return new_mat

class ElasticStiffness:

    def __init__(self,dim=3,C=[[0]]):
        self.dim = dim
        self.C = C

    def rotate(self, alpha):
        """ Rotate elasticity matrix in the 1-2 plane by an angle alpha """
        c = cos(alpha)
        s = sin(alpha)
        R = np.array([[c**2, s**2, 0, 2*s*c, 0, 0],
                      [s**2, c**2, 0, -2*s*c, 0, 0],
                      [0, 0, 1, 0, 0, 0],
                      [-c*s, c*s, 0, c**2 - s**2, 0, 0],
                      [0, 0, 0, 0,  c, s],
                      [0, 0, 0, 0, -s, c]])
        if self.dim == 2:
            R = R[np.ix_([0, 1, 3], [0, 1, 3])]
        self.C = R.dot(self.C.dot(R.T))
        self.S = R.T.dot(self.S.dot(R))

class TransverseIsotropic(ElasticStiffness):
    def __init__(self, ET, EL, nuT, nuL, muT, muL, dim=3):
```

```
        self.dim = dim
        self.ET = ET
        self.EL = EL
        self.nuT = nuT
        self.nuL = nuL
        self.muT = muT
        self.muL = muL
        Splan = np.array([[1./self.EL, -self.nuL/self.EL, -self.nuL/self.EL],
                          [-self.nuL/self.EL, 1./self.ET, -self.nuT/self.ET],
                          [-self.nuL/self.EL, -self.nuT/self.ET, 1./self.ET]])
        self.S = scipy.linalg.block_diag(Splan, np.diag([1/self.muL, 1/self.muL,
                                                         1/self.muT]))
        self.C = np.linalg.inv(self.S)

    def S_abgd(self):
        return self.S[np.ix_([0, 1, 3], [0, 1, 3])]

    def S_a3b3(self):
        return np.diag((self.S[4, 4]/4., self.S[5, 5]/4.))

    def S_ab33(self):
        return np.array([self.S[0, 2], self.S[1, 2], 0])

    def S_3333(self):
        return self.S[2, 2]


class Isotropic(TransverseIsotropic):
    def __init__(self, E, nu, dim=2):
        self.E = E
        self.nu = nu
        self.mu = self.E/2./(1+self.nu)
        self.lmbda = self.E*self.nu/(1-2*self.nu)/(1+self.nu)
        self.M = self.lmbda+2*self.mu
        TransverseIsotropic.__init__(self, E, E, nu, nu, self.mu, self.mu, dim)


mat = TransverseIsotropic(ET, EL, nuT, nuL, muT, muL)
lS_abgd = []
lS_ab33 = []
lS_a3b3 = []
lS3333 = []
lS_klmn = []
```

```
lC_klmn = []
for i in range(n):
    mat_rotate = rotate(mat, angle[i])
    lS_abgd.append(mat_rotate.S_abgd())
    lS_ab33.append(mat_rotate.S_ab33())
    lS_a3b3.append(mat_rotate.S_a3b3())
    lS3333.append(mat_rotate.S_3333())
    lS_klmn.append(mat_rotate.S)
    lC_klmn.append(mat_rotate.C)

##############################################################################
# The generalized matrix [S] of dimension  (12n - 4)  x  (12n - 4).
##############################################################################
#The generalized matrix blocks
S_k = np.zeros((6*n, 6*n))
S_knu = np.zeros((6*n, n-1))
S_kpi = np.zeros((6*n, n-1))
S_nu = np.zeros((n-1, n-1))
S_nupi = np.zeros((n-1, n-1))
S_Q = np.zeros((4*n-2, 4*n-2))
S_pi = np.zeros((n-1, n-1))

##############################################################################
# Fill of S_k
##############################################################################

for i1 in range(n):
    li = range(i1*6, (i1+1)*6)
    for j1 in range(n):
        lj = range(j1*6, (j1+1)*6)
        if li == lj:
            S_k[np.ix_(li, lj)] = np.vstack((np.hstack((lS_abgd[i1]/e[i1],
                                                        np.zeros((3, 3)))),
                                             np.hstack((np.zeros((3, 3)),
                                                 (12/(e[i1])**3)*lS_abgd[i1]))))
        else:
            continue

##############################################################################
# Fill of S_knu and S_kpi
##############################################################################
j1 = int(0)
for i1 in range(n):
```

```
    li = range(i1*6, (i1+1)*6)

    if j1 == 0:
        S_knu[np.ix_(li, [j1])] = np.vstack(((1/2)*lS_ab33[i1],
                                             (6/(5*e[i1]))*lS_ab33[i1]))
        S_kpi[np.ix_(li, [j1])] = np.vstack(((e[i1])/12*lS_ab33[i1],
                                            1/10*lS_ab33[i1]))

    elif j1 == n-1:
        S_knu[np.ix_(li, [j1-1])] = np.vstack(((1/2)*lS_ab33[i1],
                                               (-6/(5*e[i1]))*lS_ab33[i1]))
        S_kpi[np.ix_(li, [j1-1])] = np.vstack((-(e[i1])/12*lS_ab33[i1],
                                               1/10*lS_ab33[i1]))

    else:
        S_knu[np.ix_(li, [j1-1, j1])] = np.vstack((np.hstack(((1/2)*lS_ab33[i1],
                                                              (1/2)*lS_ab33[i1])),
                                                   np.hstack(((-6/(5*e[i1]))*lS_ab33[i1],
        S_kpi[np.ix_(li, [j1-1, j1])]=np.vstack((np.hstack(((-e[i1])/12*lS_ab33[i1],
                                                            (e[i1])/12*lS_ab33[i1])),
                                                 np.hstack((1/10*lS_ab33[i1],
                                                            1/10*lS_ab33[i1]))))

    j1 += 1

#############################################################################
# Fill of  S_nu,  S_pi and S_nupi
#############################################################################

for i in range(n-1):
    for j in range(n-1):
            if i == j:
                S_nu[i, j] = 13/35.*((e[i])*lS3333[i] + (e[i+1])*lS3333[i+1])
                S_nupi[i, j] = 11/210.*((e[i])**2*lS3333[i]
                                - (e[i+1])**2*lS3333[i+1])
                S_pi[i, j] = 1/105.*((e[i])**3*lS3333[i]
                             + (e[i+1])**3*lS3333[i+1])
            elif j -i == 1:
                S_nu[i, j] = S_nu[j, i] = (9/70.*(e[j]))*lS3333[j]
                S_nupi[i, j] = 13/420.*(e[j])**2*lS3333[j]
                S_nupi[j, i] = -13/420.*(e[j])**2*lS3333[j]
                S_pi[i, j] = S_pi[j, i] = -1/140.*(e[j])**3*lS3333[j]
            else:
```

```
                continue

##############################################################################
# Fill of S_Q
##############################################################################

ie = int(0)
ie1 = int(0)
ie2 = int(0)
je = int(1)
for i in range(2*n-1):
    li = range(2*i, (i+1)*2)
    for j in range(2*n-1):
        lj = range(2*j, (j+1)*2)
        if i == j and i % 2 == 0:
            S_Q[np.ix_(li, lj)] = (24/(5*e[ie]))*lS_a3b3[ie]
            ie += 1
        elif i == j:
            S_Q[np.ix_(li, lj)] = 8/15*((e[ie1])*lS_a3b3[ie1]
                                  + (e[ie1+1])*lS_a3b3[ie1+1])
            ie1 += 1
        elif abs(i-j) == 1 and j > i:
            S_Q[np.ix_(li, lj)] = S_Q[np.ix_(lj, li)] = -2/5*lS_a3b3[ie2]
            if j % 2 != 0:
                ie2 += 1
        elif abs(i - j) == 2 and i % 2 != 0 and j > i:
            S_Q[np.ix_(li, lj)] = S_Q[np.ix_(lj, li)] = -2/15*(e[je])*lS_a3b3[je]
            je += 1
        else:
            continue

S_F = np.vstack((np.hstack((S_k, S_knu, S_kpi)),
                 np.hstack((S_knu.T, S_nu, S_nupi)),
                 np.hstack((S_kpi.T, S_nupi.T, S_pi))))

IS_F and IS_C are the inerve matrix of S_F and S_Q respectively
IS_F = np.linalg.inv(S_F)
IS_C = np.linalg.inv(S_Q)

##############################################################################
# The orthogonal Legendre-like polynomial basis defined on layer i.
We used later in the construction of the 3D stress fields σ_kl^3D.
##############################################################################
```

```
def base(z):
    P0 = []
    P1 = []
    P2 = []
    P3 = []
    for i in range(n):
        P0.append(1)
        P1.append((z-mh[i])/e[i])
        P2.append(-6*((z-mh[i])/e[i])**2 + 1./2)
        P3.append(-2*((z-mh[i])/e[i])**3 + 3/10 * ((z-mh[i])/e[i]))
    return (P0, P1, P2, P3)



##########################################################################
# The linear and quadratic hat functions and their derivative functions.
We use those functions in the reconstruction of the 3D displacement fields
##########################################################################

def N(z):
    N1 = []
    N2 = []
    for i in range(n):
        N1.append((hb[i+1] - z)/e[i])
        N1.append((z-hb[i])/e[i])
        N2.append((2/(e[i]**2)) * (z - mh[i])*(z - hb[i+1]))
        N2.append((-4/(e[i]**2))*(z - hb[i])*(z - hb[i+1]))
        N2.append((2/(e[i]**2))*(z - mh[i])*(z - hb[i]))
    return (N1, N2)


def deriveN(z):
    DN1 = []
    DN2 = []
    for i in range(n):
        DN1.append(-1/e[i])
        DN1.append(1/e[i])
        DN2.append((2/(e[i]**2))*(2*z - mh[i] - hb[i+1]))
        DN2.append((-4/(e[i]**2))*(2*z - hb[i] - hb[i+1]))
        DN2.append((2/(e[i]**2))*(2*z - mh[i] - hb[i]))

    return(DN1, DN2)
```

```
while iteration < MAX_ITER:
    mesh = new_mesh

    ########################################################################
    # The Functional space for the generalized displacements
    ########################################################################

    U_3 = FiniteElement("CG", mesh.ufl_cell(), 2)
    U_alpha = VectorElement("CG", mesh.ufl_cell(), 2)
    Phi_alpha = VectorElement("CG", mesh.ufl_cell(), 2)
    Vi = FiniteElement("CG", mesh.ufl_cell(), 2)

    list_3 = []
    for i in range(n):
        list_3.append(U_alpha)
        list_3.append(U_3)
        list_3.append(Phi_alpha)
    for i in range(n-1):
        list_3.append(Vi)

    V = FunctionSpace(mesh, MixedElement(list_3))
    VD = VectorFunctionSpace(mesh, 'Lagrange', 1)
    P1 = FunctionSpace(mesh, 'CG', 1)  # pour le gradient
    P2 = VectorFunctionSpace(mesh, 'CG', 1)
    P3 = TensorFunctionSpace(mesh, 'CG', 1)
    P1_d = FunctionSpace(mesh, 'DG', 0)
    P2_d = VectorFunctionSpace(mesh, 'DG', 0)
    P3_d = TensorFunctionSpace(mesh, 'DG', 0)
```

To interpolate the values of load on the meshe we do the following :

```
    f1 = interpolate(Expression("l", l=load[0], element=P1._ufl_element), P1)
    f2 = interpolate(Expression("l", l=load[1], element=P1._ufl_element), P1)
    f3 = interpolate(Expression("l", l=load[2], element=P1._ufl_element), P1)
    f4 = interpolate(Expression("l", l=load[3], element=P1._ufl_element), P1)

    ########################################################################
    # Limits conditions
    ########################################################################

    def border(x, on_boundary):
        return on_boundary
```

We consider the plates totally clamped for the Homogeneous and Triple laminated

```
j = 1
bc = []
for i in range(n):
   bc += [DirichletBC(V.sub(iUa(n)[i]), Constant((0.,0.)), border),
          DirichletBC(V.sub(iP(n)[i]), Constant((0.,0.)), border),
          DirichletBC(V.sub(iU3(n)[i]), Constant(0.), border),
          DirichletBC(V.sub(iV(n)[j-1]), Constant(0.), border)]
   j += 1
```

For the laminate with a circular hole, a tensile loading is applied to the plate through an imposed horizontal displacement $U^i = \pm Ue_x$ *for all plies* i

```
def left(x, on_boundary):
    return near(x[0], -L/2) and on_boundary

def right(x, on_boundary):
    return near(x[0], L/2) and on_boundary

bc = []
for i in range(n):
    bc += [DirichletBC(V.sub(iUa(n)[i]).sub(0), Constant(-u_t), left),
           DirichletBC(V.sub(iUa(n)[i]).sub(0), Constant(u_t), right),
           DirichletBC(V.sub(iUa(n)[i]).sub(1), Constant(0.), left),
           DirichletBC(V.sub(iUa(n)[i]).sub(1), Constant(0.), right),
           DirichletBC(V.sub(iU3(n)[i]), Constant(0.), left),
           DirichletBC(V.sub(iU3(n)[i]), Constant(0.), right)]
```

For the DCB with delaminated interface, the plate is clamped on its right boundary, and positive (resp. negative) displacements $U_3^i = +U$ *(resp.* $U_3^i = -U$*) are enforced on the left part for the top layers* $i = 4;\ 5;\ 6$ *(resp. bottom layers* $i = 1;\ 2;\ 3$*)*

```
 def left(x, on_boundary):
    return near(x[0], 0.) and on_boundary
def right(x, on_boundary):
    return near(x[0], L) and on_boundary
u_t = 1
bc = []
for i in range(3):
    bc += [DirichletBC(V.sub(iU3(n)[i]), Constant(-u_t), left),
           DirichletBC(V.sub(iU3(n)[i+3]), Constant(u_t), left)]
```

```
for i in range(n):
    bc += [DirichletBC(V.sub(iUa(n)[i]), Constant((0.,0.)), right),
           DirichletBC(V.sub(iP(n)[i]), Constant((0.,0.)), right),
           DirichletBC(V.sub(iU3(n)[i]), Constant(0.), right)]

for j in range(n-1):
    bc+= [DirichletBC(V.sub(iV(n)[j-1]), Constant(0.), right)]


#############################################################################
# The generalized displacement vector [δ]
#############################################################################

def generalized_displ(x):
    list_u = []
    Dg = split(x)
    for i in range(n):
        list_u.append(Dg[iUa(n)[i]][0])
        list_u.append(Dg[iUa(n)[i]][1])
        list_u.append(Dg[iU3(n)[i]])
        list_u.append(Dg[iP(n)[i]][0])
        list_u.append(Dg[iP(n)[i]][1])
    for j in range(n-1):
        list_u.append(Dg[iV(n)[j]])
    return as_vector(list_u)

#############################################################################
# Function that give us the deformations tensor
#############################################################################

def generalized_deformation(x):
    list_epsilon_Chi = []
    list_D_nu = []
    list_Gamma_D = []
    list_lambda = []
    DG = split(x)
    for i in range(n):
        list_epsilon_Chi.append(sym(grad(DG[iUa(n)[i]])))
        list_epsilon_Chi.append(sym(grad(DG[iP(n)[i]])))
        list_Gamma_D.append(DG[iP(n)[i]] + grad(DG[iU3(n)[i]]))
        if i < n-1:
            list_Gamma_D.append(DG[iUa(n)[i+1]] - DG[iUa(n)[i]]
                                -e[i]/2 * DG[iP(n)[i]] -e[i+1]/2 *DG[iP(n)[i+1]]
```

```
                                + grad(DG[iV(n)[i]]))
            list_D_nu.append(DG[iU3(n)[i+1]] - DG[iU3(n)[i]])
            list_lambda.append(DG[iV(n)[i]])
    return (list_epsilon_Chi, list_D_nu, list_Gamma_D, list_lambda)



########################################################################
# function that convert the deformation tensor into deformation vector: [E]
########################################################################
def deformation2voigt(x):
    list_epsilon_Chi, list_D_nu, list_Gamma_D, list_lambda = x
    list_E = []
    list_G = []

    for i in range(2*n):
        list_E.append(list_epsilon_Chi[i][0,0])
        list_E.append(list_epsilon_Chi[i][1,1])
        list_E.append(2*list_epsilon_Chi[i][0,1])
    for j in range(n):
        list_G.append(list_Gamma_D[2*j][0])
        list_G.append(list_Gamma_D[2*j][1])
        if j < n-1:
            list_G.append(list_Gamma_D[2*j+1][0])
            list_G.append(list_Gamma_D[2*j+1][1])

    list_total_deformation = list_E + list_D_nu + list_lambda
    return (as_vector(list_total_deformation), as_vector(list_G))



########################################################################
# The function that give us the values of [S]^-1[E]
########################################################################

def generalized_stress(x):
    C = [as_tensor(IS_F), as_tensor(IS_C)]
    EE = deformation2voigt(generalized_deformation(x))
    return dot(C[0], EE[0]), dot(C[1], EE[1])


########################################################################
# function to calculate the generalized prestrain vector [E^P]
```

```
#########################################################################

def EP(load):
    Ep_ec = []
    Ep_dnu = []
    Ep_l = []
    Ep_Gd = []
    Z3 = [0]*(4*n - 10)
    Z1 = [0]*(6*n - 12)
    Z2 = [0]*(n - 3)
    pim = grad(f2)[0]+grad(f4)[1]
    pip = grad(f1)[0]+grad(f3)[1]
    t = as_vector(lS_ab33[0][:,0])*(-1/2*(load[5])+(e[0])/12*(pim))
    Ep_ec += [t[0], t[1], t[2]]
    t = as_vector(lS_ab33[0][:,0])*(6/(5*(e[0]))*(load[5]) - 1/10*(pim))
    Ep_ec += [t[0], t[1], t[2]]
    Ep_ec += Z1
    t = as_vector(lS_ab33[n-1][:,0])*(1/2*(load[4]) + (e[n-1])/12*(pip))
    Ep_ec += [t[0], t[1], t[2]]
    t = as_vector(lS_ab33[n-1][:,0])*(6/(5*(e[n-1]))*(load[4])+1/10*(pip))
    Ep_ec += [t[0], t[1], t[2]]

    Ep_Gd += (2/5*np.dot(lS_a3b3[0], np.array([load[1], load[3]]))).tolist()

    if n > 2:
        Ep_dnu.append(-9/70*(e[0])*lS3333[0]*(load[5])
                      + 13/420*(e[0])**2*lS3333[0]*(pim))
        Ep_dnu += Z2
        Ep_dnu.append(9/70*(e[n-1])*(lS3333[n-1])*(load[4])
                      + 13/420*(e[n-1])**2*(lS3333[n-1])*(pip))
        Ep_l.append(-13/420*(e[0])**2*lS3333[0]*(load[5])
                    + 1/140*(e[0])**3*lS3333[0]*(pim))
        Ep_l += Z2
        Ep_l.append(-13/420*(e[n-1])**2*lS3333[n-1]*(load[4])
                    - 1/140*(e[n-1])**3*lS3333[n-1]*(pip))

        Ep_Gd += ((2/15)*e[0]*np.dot(lS_a3b3[0],
                  np.array([load[1] , load[3]]))).tolist()
        Ep_Gd += Z3
        Ep_Gd += (-(2/15)*(e[n-1])*np.dot(lS_a3b3[n-1],
                  np.array([load[0] , load[2]]))).tolist()

    elif n == 2:
```

```
        Ep_dnu.append(-9/70*(e[0])*lS3333[0]*(load[5])
                      + 13/420*(e[0])**2*lS3333[0]*(pim)
                      + 9/70*(e[n-1])*(lS3333[n-1])*(load[4])
                      + 13/420*(e[n-1])**2*(lS3333[n-1])*(pip))
        Ep_l.append(-13/420*(e[0])**2*lS3333[0]*(load[5])
                    + 1/140*(e[0])**3*lS3333[0]*(pim)
                    - 13/420*(e[n-1])**2*lS3333[n-1]*(load[4])
                    -1/140*(e[n-1])**3*lS3333[n-1]*(pip))

        Ep_Gd += ((2/15)*(e[0])*np.dot(lS_a3b3[0],
                                       np.array([load[1] , load[3]]))
                - (2/15)*(e[n-1])*np.dot(lS_a3b3[1],
                                       np.array([load[0] , load[2]]))).tolist()

    Ep_Gd += (-2/5*np.dot(lS_a3b3[n-1],
                          np.array([load[0], load[2]]))).tolist()

    Ep = Ep_ec + Ep_dnu + Ep_l
    return as_vector(Ep), as_vector(Ep_Gd)



##########################################################################
# Function to calculate the generalized forces vector F
##########################################################################

def generalized_forces(load):
    F = []
    Z = [0]*(n-1)
    Z1 = [0]*(5*n -10)
    F.append(load[1])
    F.append(load[3])
    F.append(load[5])
    F.append(-(e[0])/2*(load[1]))
    F.append(-(e[0])/2*(load[3]))
    F += Z1
    F.append((load[0]))
    F.append((load[2]))
    F.append((load[4]))
    F.append(e[n-1]/2*(load[0]))
    F.append(e[n-1]/2*(load[2]))
    F = F + Z
    return as_vector(F)
```

```
#########################################################################
#Function that calculate the generalized stresses [Σ]
Where [Σ] = [S]^-1 ([E] - [E^P])
#########################################################################

def generalized_STRESS(x):
    C = [as_tensor(IS_F), as_tensor(IS_C)]
    EE = deformation2voigt(generalized_deformation(x))
    return dot(C[0], EE[0] - EP(load)[0]), dot(C[1], EE[1] - EP(load)[1])

#########################################################################
# Solver of linear and bi-linear forms
#########################################################################

u = Function(V)
DG = split(u)
u_ = TestFunction(V)
du = TrialFunction(V)

We used dx_shear to provide the shear locking
dx_shear = dx(metadata="quadrature_degree": 2)

l = dot(generalized_stress(u_)[0], EP(load)[0])*dx
    +dot(generalized_forces(load), generalized_displ(u_))*dx
a = inner(generalized_stress(u_)[0], deformation2voigt(
                    generalized_deformation(du))[0])*dx
+ inner(generalized_stress(u_)[1], deformation2voigt(
                    generalized_deformation(du))[1])*dx_shear

solve(a == l, u, bc, solver_parameters="linear_solver": "mumps")

#########################################################################
# Extract the values of the generalized stresses as a lists
#########################################################################
Sig_F = generalized_STRESS(u)[0]
Sig_C = generalized_STRESS(u)[1]
List_N = []
List_M = []
List_Q = []
List_tau = []
List_pi = []
```

```
List_nu = []
ln = 1
lm = 1


def voigt2NM(S, i):
    return as_tensor([[S[3*i], S[3*i + 2]], [S[3*i + 2], S[3*i + 1]]])


def voigt2n(S, j):
    return S[6*n + j]


def voigt2p(S, j):
    return S[7*n - 1 + j]


def voigt2QT(S, i):
    return as_vector([S[2*i], S[2*i + 1]])


for i in range(2*n):
   if i % 2 == 0:
       Ni = Function(P3, name = "N - layer "+str(ln))
       Ni.assign(project(voigt2NM(Sig_F, i), P3_d))
       List_N.append(Ni)


       Qi = Function(P2, name = "Q - layer"+str(ln))
       Qi.assign(project(voigt2QT(Sig_C, i), P2_d))
       List_Q.append(Qi)
       ln+= 1
   else:
       Mi = Function(P3, name = "M - layer "+str(lm))
       Mi.assign(project(voigt2NM(Sig_F, i), P3_d))
       List_M.append(Mi)
       if i < 2*n - 1:
           Ti = Function(P2, name = "Tau - interfaces"+str(lm) + str(lm+1))
           Ti.assign(project(voigt2QT(Sig_C, i), P2_d))
           List_tau.append(Ti)
       lm+= 1
```

```
for j in range(n-1):
    ni = Function(P1, name = "Nu- interfaces"+str(j+1) + str(j+2))
    ni.assign(project(voigt2n(Sig_F, j), P1_d))
    List_nu.append(ni)

    PI = Function(P1, name = "Pi - interfaces"+str(j+1) + str(j+2))
    PI.assign(project(voigt2p(Sig_F, j), P1_d))
    List_pi.append(PI)

##############################################################################
# Function to calculate the 3D stress field σ
##############################################################################

def sigma_3D(z, con):
    tauzm = [-load[1], -load[3]]
    taunp = [load[0], load[2]]
    nuzm = - load[5]
    nunp = load[4]
    pim = grad(f2)[0]+grad(f4)[1]
    pip = grad(f1)[0]+grad(f3)[1]


    sig_ab = (base(z)[0][con - 1]/e[con - 1])*List_N[con-1]
            + 12/(e[con - 1])**2 *base(z)[1][con - 1]* List_M[con - 1]

    if con > 1 and con < n:
        sig_a3 = List_Q[con - 1]*base(z)[0][con - 1]/e[con - 1]
                 + (List_tau[con - 1] - List_tau[con - 2])*base(z)[1][con - 1]
                 + (List_Q[con - 1] - e[con - 1]/2*(List_tau[con - 1]
                 + List_tau[con - 2]))*base(z)[2][con - 1]/e[con - 1]
        sig_33 = (1/2 * (List_nu[con - 1] + List_nu[con - 2])
                 + e[con - 1]/12 * (List_pi[con - 1]
                    - List_pi[con - 2]))*base(z)[0][con - 1]
                  + (e[con - 1]/10 * (List_pi[con - 1] + List_pi[con - 2])
                    + 6/5 * (List_nu[con - 1]
                    - List_nu[con - 2]))*base(z)[1][con - 1]
                  + e[con - 1]/12 * (List_pi[con- 1]
                    - List_pi[con - 2]) * base(z)[2][con - 1]
                  + (e[con - 1]/2 * (List_pi[con - 1] + List_pi[con - 2])
                    + (List_nu[con - 1] - List_nu[con - 2]))*base(z)[3][con - 1]
    elif con == 1:
        sig_a3 = List_Q[con - 1]*base(z)[0][con - 1]/e[con - 1]
                + (List_tau[con - 1] - as_vector(tauzm))*base(z)[1][con - 1]
```

```
                + (List_Q[con - 1] - e[con - 1]/2*(List_tau[con - 1]
                + as_vector(tauzm)))*base(z)[2][con - 1]/e[con - 1]
        sig_33 = (1/2 * (List_nu[con - 1] + nuzm)
                + e[con - 1]/12 * (List_pi[con - 1] + pim))*base(z)[0][con - 1]
                + (e[con - 1]/10 * (List_pi[con - 1] - pim)
                + 6/5 * (List_nu[con - 1] - nuzm))*base(z)[1][con - 1]
                + e[con - 1]/12 * (List_pi[con -1] + pim) * base(z)[2][con - 1]
                + (e[con - 1]/2 * (List_pi[con - 1] - pim)
                + (List_nu[con - 1] - nuzm))*base(z)[3][con - 1]
    elif con == n:
        sig_a3 = List_Q[con - 1]*base(z)[0][con - 1]/e[con - 1]
                + (as_vector(taunp) - List_tau[con - 2])*base(z)[1][con - 1]
                + (List_Q[con - 1] - e[con - 1]/2*(as_vector(taunp)
                   + List_tau[con - 2]))*base(z)[2][con - 1]/e[con - 1]
        sig_33 = (1/2 * (nunp + List_nu[con - 2])
                + e[con - 1]/12 * (pip - List_pi[con - 2]))*base(z)[0][con - 1]
                + (e[con-1]/10 * (pip + List_pi[con - 2])
                + 6/5 * (nunp - List_nu[con - 2]))*base(z)[1][con - 1]
                + e[con - 1]/12 * (pip - List_pi[con - 2]) * base(z)[2][con - 1]
                + (e[con - 1]/2 * (pip + List_pi[con - 2])
                + (nunp - List_nu[con - 2]))*base(z)[3][con - 1]
    return (sig_ab[0,0], sig_ab[1,1], sig_33, sig_ab[0,1], sig_a3[0], sig_a3[1])

##############################################################################
# Here we calculate the reconstructed displacement
##############################################################################
```

Calculate of U_{alpha} :

```
A = np.zeros((n+1, n+1))
for i in range(n+1):
    for j in range(n+1):
        if i == j:
            if i != 0 and i != n:
                A[i, j] = (e[i-1] + e[i])/3
            elif i == 0:
                A[i, j] = e[0]/3
            elif i == n:
                A[i, j] = e[n-1]/3
        elif j - i == 1:
            A[i, j] = A[j, i] = e[i]/6
A[0,:] = 0
A[0,0] = 1
A[-1,:] = 0
```

```
A[-1,-1] =1

A1 = np.kron(A, np.eye(2))
IA = np.linalg.inv(A1)


def Fa(x):
    list_Fa = []
    Dg = split(x)
    t = as_vector(Dg[iUa(n)[0]] - e[0]/2 * Dg[iP(n)[0]])
    list_Fa += [t[0], t[1]]
    for i in range(n-1):
        t = as_vector(e[i]/2 * Dg[iUa(n)[i]] + e[i+1]/2 * Dg[iUa(n)[i+1]]
                      + (e[i]**2)/12 * Dg[iP(n)[i]]
                      - (e[i+1]**2)/12 * Dg[iP(n)[i+1]])
        list_Fa += [t[0], t[1]]
    t = as_vector(Dg[iUa(n)[n-1]] + e[n-1]/2 * Dg[iP(n)[n-1]])
    list_Fa += [t[0], t[1]]
    return as_vector(list_Fa)


def deplacement_alpha_Chapeau(x):
    return dot(as_tensor(IA), Fa(x))
```

Calculate of U_3 :

```
B2 = np.zeros((2*n-1, n+1))
indice = 0
c = 0
for i in range(2*n -1):
    if i % 2 == 0:
        i1 = indice
        i2 =indice + 1
        B2[i, i1] = B2[i, i2] = 1/2
    if i % 2 != 0:
        i1 = indice
        i2 =indice + 1
        i3 =indice + 2
        B2[i, i1] = -e[c]/12
        B2[i, i2] = (e[c]+e[c+1])/12
        B2[i, i3] = -e[c+1]/12
        indice += 1
        c +=1
```

```
B2T = np.transpose(B2)
MB = B2T.dot(B2)
B2F = np.linalg.inv(MB)

def Fb(x):
    list_Fb = []
    Dg = split(x)
    for i in range(n-1):
        list_Fb.append(Dg[iU3(n)[i]])
        list_Fb.append(Dg[iV(n)[i]])
    list_Fb.append(Dg[iU3(n)[n-1]])
    return as_vector(list_Fb)


def deplacement_3_Chapeau(x):
    return dot(as_tensor(B2F), dot(as_tensor(B2T), Fb(x)))


##########################################################################
# Construction of the reconstructed deformation fields epsilon_kl
##########################################################################

def epsilon_alpbeta(z, con):

    eps_ab = sym(grad(vecQa(dep_a, con)))*N(z)[0][2*(con - 1)]
            + sym(grad(vecQa(dep_a, con + 1)))*N(z)[0][2*(con - 1) + 1]

    eps_33 = deplacement_3_Chapeau(u)[con -1] * deriveN(z)[0][2*(con - 1)]
            + deplacement_3_Chapeau(u)[con]*deriveN(z)[0][2*(con - 1) + 1]

    eps_a3 = 1/2*(vecQa(dep_a, con) * deriveN(z)[0][2*(con - 1)]
                  + vecQa(dep_a, con + 1) * deriveN(z)[0][2*(con - 1) + 1]
                  + grad(deplacement_3_Chapeau(u)[con - 1])
                  * N(z)[0][2*(con - 1)] + grad(deplacement_3_Chapeau(u)[con])
                  * N(z)[0][2*(con - 1) + 1])
    return eps_ab[0, 0], eps_ab[1, 1], eps_33,
           2*eps_ab[0, 1], 2*eps_a3[0], 2*eps_a3[1]


##########################################################################
# Calculate of the reconstructed stress field
##########################################################################
```

```
def sigma_chap(z, con):
    C1 = as_tensor(lC_klmn[con - 1])
    P = dot(C1, as_vector(epsilon_alpbeta(z, con)))
    return P

###############################################################################
# The Re-mesh strategy and the estimator error
###############################################################################

poids = [(18 + sqrt(30))/36, (18 - sqrt(30))/36]
zg = [-sqrt(3/7 - 2/7 * sqrt(6/5)), sqrt(3/7 - 2/7 * sqrt(6/5)),
      -sqrt(3/7 + 2/7 * sqrt(6/5)), sqrt(3/7 + 2/7 * sqrt(6/5))]

# function to calculate the estimator error
def estimateurerreur(poids, zg):
    estimateur = 0
    for i in range(n):
        S = as_tensor(lS_klmn[i])
        p = 0
        Sum = 0
        for g in range(4):
            diff = as_vector(sigma_3D(e[i]/2 * zg[g] + mh[i], i+1))
                  - as_vector(sigma_chap(e[i]/2 * zg[g] + mh[i], i+1))
            Sum += poids[p] * inner(diff, dot(S,diff))
            if g % 2 != 0:
                p += 1
        estimateur += (e[i]/4 * Sum)
    return estimateur



K = CellVolume(mesh)
cell_markers = MeshFunction("bool", mesh, mesh.topology().dim())
V0 = FunctionSpace(mesh, "DG", 0)
gamma = project(K*estimateurerreur(poids, zg), V0).vector().get_local()
index_sort = np.argsort(gamma)[::-1]
cum_sum = np.cumsum(gamma[index_sort])
for c in cells(mesh):
    cell_markers[c] = cum_sum[np.where(c.index()==index_sort)]
                      < REFINE_RATIO*sum(gamma)

new_mesh = refine(mesh, cell_markers)
```

```
    iteration +=1
```

3 Annex C

In this Annex we presents the code that apply and solve the hybridization of a mixed method for the SCLS1 using Firedrake.
Firedrake is a new tool for automating the numerical solution of partial differential equations. Firedrake adopts the domain-specific language for the finite element method of the FEniCS project, but with a pure Python runtime-only implementation centered on the composition of several existing and new abstractions for particular aspects of scientific computing. The result is a more complete separation of concerns that eases the incorporation of separate contributions from computer scientists, numerical analysts, and application specialists. These contributions may add functionality or improve performance.

Firedrake benefits from automatically applying new optimizations. This includes factorizing mixed function spaces, transforming and vectorizing inner loops, and intrinsically supporting block matrix operations. Importantly, Firedrake presents a simple public API for escaping the UFL abstraction. This allows users to implement common operations that fall outside of pure variational formulations, such as flux limiters.
As we show in 3.7, we have 3 illustration examples. In the code below we show the modeling of each examples.

```
######################################################################
# For the Homogeneous laminate
######################################################################
# interpolation degree
interp = 1

# Nombre of layer
n = 2

# The mesh
L = 1
Ne = 70
assert (Ne % 2 == 0), "Number of elements should be even"
subprocess.call(["gmsh", "-2", "-format", "msh2", "-setnumber", "N", str(Ne),
                 "square_plate.geo"])
mesh = Mesh("square_plate.msh")

# The thickness
```

```
h = 0.01

#The constittive materials
E = 10.
nu = 0.3
mu = E/2./(1+nu)

######################################################################
# For the laminate with a circular hole
######################################################################
# element sizes
dplate = 0.1
dhole = 0.025

# Radius
R = 0.15

# interpolation degree
interp = 2

# The mesh
subprocess.call(["gmsh", "-2", "-format", "msh2", "-setnumber", "d", str(dplate),
                 "-setnumber", "d2", str(dhole), "-setnumber", "R", str(R),
                 "plate_hole.geo"])

logging.set_level(DEBUG)

mesh = Mesh("plate_hole.msh")

#Fiber orientation angles
angle = [90*pi/180.,  45*pi/180., 0*pi/180., 45*pi/180., 90*pi/180.]

# Nombre of layer
n = len(angle)

#The constittive materials
EL = 137.9e3
ET = 14.48e3
nuL = 0.21
nuT = 0.21
muL = 5.86e3
muT = 5.86e3
######################################################################
```

```
# For the Bending of a laminate with multi-cracking
#####################################################################
# Number of elements along x and y
nx = 35
ny = 5

#The mesh
subprocess.call(["gmsh", "-2", "-format", "msh2", "-setnumber", "nx", str(nx+1),
                 "-setnumber", "ny", str(ny+1), "plate_cracks.geo"])

mesh = Mesh("plate_cracks.msh")

#interpolation degree
interp = 2

#orientation angles
angle = [0*pi/180.,  90*pi/180., 0*pi/180, 90*pi/180., 0*pi/180.]

#Nomber of layers
n = len(angle)

#The constittive materials
EL = 137.9e3
ET = 14.48e3
nuL = 0.21
nuT = 0.21
muL = 5.86e3
muT = 5.86e3
```

Note that we used the same functions in Annex B to discritize the thickness and to calculate the values of h_i^-, h_i^+ *and* $\overline{h}_i$. *Also we used the same functions as in the annexe B to determinate the flexibility, the generalized matrix and the function that give us the orthogonal bases and the linear and quadratique hat functions and their derivatives*

```
#####################################################################
# Functional Spaces for the generalized stresses, displacement,
and the Lagrange multipliers to the three illustrative exepmles
#####################################################################

C_e = VectorFunctionSpace(mesh, "DG", interp, dim=12*n-4)
D_e = VectorFunctionSpace(mesh, "DG", interp-1, dim=6*n-1)
```

```
M_e = VectorFunctionSpace(mesh, "Discontinuous Lagrange Trace", interp, dim=6*n-1)
V = C_e * D_e * M_e

######################################################################
# Functions that gives us the values of the generalized stresses,
displacements and the Lagrange multipliers
######################################################################

def N_alphabeta(x):
    Dg = split(x)
    List_N = []
    for i in range(n):
        List_N1 = []
        List_N1.append(Dg[0][6*i])
        List_N1.append(Dg[0][6*i + 1])
        List_N1.append(Dg[0][6*i + 2])
        List_N.append(as_vector(List_N1))
    return List_N

def M_alphabeta(x):
    Dg = split(x)
    List_M = []
    for i in range(n):
        List_M1 = []
        List_M1.append(Dg[0][6*i + 3])
        List_M1.append(Dg[0][6*i + 4])
        List_M1.append(Dg[0][6*i + 5])
        List_M.append(as_vector(List_M1))
    return List_M

def nu(x):
    Dg = split(x)
    List_nu = []
    for j in range(n-1):
        List_nu.append(Dg[0][6*n + j])
    return List_nu

def Q_alpha(x):
    Dg = split(x)
    List_Q = []
    for i in range(n):
        List_Q1 = []
        List_Q1.append(Dg[0][7*n - 1 + 4*i])
```

```
        List_Q1.append(Dg[0][7*n -1 + 4*i + 1])
        List_Q.append(as_vector(List_Q1))
    return List_Q

def tau_alpha(x):
    Dg = split(x)
    List_tau = []
    for j in range(n-1):
        List_tau1 = []
        List_tau1.append(Dg[0][7*n - 1 + 4*j + 2])
        List_tau1.append(Dg[0][7*n - 1 + 4*j + 3])
        List_tau.append(as_vector(List_tau1))
    return List_tau

def pi(x):
    Dg = split(x)
    List_pi = []
    for j in range(n-1):
        List_pi.append(Dg[0][11*n - 3 + j])
    return List_pi

def u_alpha(x):
    Dg = split(x)
    List_u =[]
    for i in range(n):
        List_u1 = []
        List_u1.append(Dg[1][5*i])
        List_u1.append(Dg[1][5*i + 1])
        List_u.append(as_vector(List_u1))
    return List_u

def u_3(x):
    Dg = split(x)
    List_u3 = []
    for i in range(n):
        List_u3.append(Dg[1][5*i + 2])
    return List_u3

def phi_alpha(x):
    Dg = split(x)
    List_phi = []
    for i in range(n):
        List_phi1 = []
```

```
        List_phi1.append(Dg[1][5*i + 3])
        List_phi1.append(Dg[1][5*i + 4])
        List_phi.append(as_vector(List_phi1))
    return List_phi

def V_j(x):
    Dg = split(x)
    List_v = []
    for j in range(n-1):
        List_v.append(Dg[1][5*n + j])
    return List_v

def v_N_alpha(x):
    Dg = split(x)
    List_vN_alpha =[]
    for i in range(n):
        List_vN_alpha2 = []
        List_vN_alpha2.append(Dg[2][4*i])
        List_vN_alpha2.append(Dg[2][4*i + 1])
        List_vN_alpha.append(as_vector(List_vN_alpha2))
    return List_vN_alpha

def v_M_alpha(x):
    Dg = split(x)
    List_vM_alpha =[]
    for i in range(n):
        List_vM_alpha2 = []
        List_vM_alpha2.append(Dg[2][4*i + 2])
        List_vM_alpha2.append(Dg[2][4*i + 3])
        List_vM_alpha.append(as_vector(List_vM_alpha2))
    return List_vM_alpha

def w_q(x):
    Dg = split(x)
    List_wq =[]
    for i in range(n):
       List_wq.append(Dg[2][4*n + 2*i])
    return List_wq

def w_t(x):
    Dg = split(x)
    List_wt = []
    for j in range(n-1):
```

```
        List_wt.append(Dg[2][4*n + 2*j + 1])
    return List_wt

######################################################################
# Function that organise the generalized displacements as a list
######################################################################

def generalized_displ(x):
    list_u = []
    for i in range(n):
        list_u.append(u_alpha(x)[i][0])
        list_u.append(u_alpha(x)[i][1])
        list_u.append(u_3(x)[i])
        list_u.append(phi_alpha(x)[i][0])
        list_u.append(phi_alpha(x)[i][1])
    for j in range(n-1):
        list_u.append(V_j(x)[j])
    return list_u

######################################################################
# Function that organise generalized stresses as a list
######################################################################

def generalized_stress(x):
    list_NM = []
    list_nu = []
    list_QT = []
    list_pi = []
    for i in range(n):
        list_NM.append(N_alphabeta(x)[i][0])
        list_NM.append(N_alphabeta(x)[i][1])
        list_NM.append(N_alphabeta(x)[i][2])
        list_NM.append(M_alphabeta(x)[i][0])
        list_NM.append(M_alphabeta(x)[i][1])
        list_NM.append(M_alphabeta(x)[i][2])
        list_QT.append(Q_alpha(x)[i][0])
        list_QT.append(Q_alpha(x)[i][1])
        if i<n-1:
            list_QT.append(tau_alpha(x)[i][0])
            list_QT.append(tau_alpha(x)[i][1])
    for j in range(n-1):
        list_nu.append(nu(x)[j])
        list_pi.append(pi(x)[j])
```

```
    list_total_stress = list_NM + list_nu +  list_QT + list_pi
    return list_total_stress

######################################################################
# Function that organise the Lagrange multipliers as a list
######################################################################

def multplicateur_Lagrange(x):
    List_VNM = []
    List_VQT = []
    for i in range(n):
        List_VNM.append(v_N_alpha(x)[i][0])
        List_VNM.append(v_N_alpha(x)[i][1])
        List_VNM.append(v_M_alpha(x)[i][0])
        List_VNM.append(v_M_alpha(x)[i][1])
        List_VQT.append(w_q(x)[i])
        if i <n-1:
            List_VQT.append(w_t(x)[i])
    list_total_lagrange = List_VNM + List_VQT
    return as_vector(list_total_lagrange)
```

In addition, we uses the same function as before to calculate $[E^P]$ *and* $[F]$

```
######################################################################
# Function that convert from vector to tensor
######################################################################

def totensor(x):
    return as_tensor([[x[0], x[2]], [x[2], x[1]]])

######################################################################
# Functions that calculate the derivatives of the generalized stresses
N, M, Q, τ
######################################################################

def DN(x):
    List_DN = []
    for i in range(n):
        List_DN1 = []
        List_DN1.append(div(totensor(N_alphabeta(x)[i]))[0])
        List_DN1.append(div(totensor(N_alphabeta(x)[i]))[1])
```

```
        List_DN.append(as_vector(List_DN1))
    return List_DN


def DM(x):
    List_DM = []
    for i in range(n):
        List_DM1 = []
        List_DM1.append(div(totensor(M_alphabeta(x)[i]))[0])
        List_DM1.append(div(totensor(M_alphabeta(x)[i]))[1])
        List_DM.append(as_vector(List_DM1))
    return List_DM


def DQ(x):
    List_DQ = []
    for i in range(n):
        List_DQ.append(div(Q_alpha(x)[i]))
    return List_DQ

def DT(x):
    List_DT = []
    for i in range(n-1):
        List_DT.append(div(tau_alpha(x)[i]))
    return List_DT

######################################################################
# Function that present the equilibrium equations 1.11.
and second a function that convert the equilibrium equations into list
######################################################################

def EDE1(x):
    E1 = [0]*n
    E2 = [0]*n
    E3 = [0]*n
    E4 = [0]*(n-1)

    E1[0] = DN(x)[0] + tau_alpha(x)[0]
    E2[0] = DQ(x)[0] + nu(x)[0]
    E3[0] = DM(x)[0] - Q_alpha(x)[0] + (e[0]/2) * (tau_alpha(x)[0])
    E4[0] = DT(x)[0] - pi(x)[0]
    for i in range(1, n-1):
        E1[i] = DN(x)[i] + tau_alpha(x)[i] - tau_alpha(x)[i-1]
```

```
        E2[i] = DQ(x)[i] + nu(x)[i] - nu(x)[i-1]
        E3[i] = DM(x)[i] - Q_alpha(x)[i] + (e[i]/2)*(tau_alpha(x)[i] + tau_alpha(x)[i
        E4[i] = DT(x)[i] - pi(x)[i]
    E1[n-1] = DN(x)[n-1] - tau_alpha(x)[n-2]
    E2[n-1] = DQ(x)[n-1] - nu(x)[n-2]
    E3[n-1] = DM(x)[n-1] - Q_alpha(x)[n-1] + e[n-1]/2 *(tau_alpha(x)[n-2])

    return (E1, E2, E3, E4)

def EDE_tovec(x):
    List = [0]*(6*n - 1)
    for i in range(n):
        List[5*i] = EDE1(x)[0][i][0]
        List[5*i + 1] = EDE1(x)[0][i][1]
        List[5*i + 2] = EDE1(x)[1][i]
        List[5*i + 3] = EDE1(x)[2][i][0]
        List[5*i + 4] = EDE1(x)[2][i][1]
    for j in range(n-1):
        List[5*n + j] = EDE1(x)[3][j]
    return List

######################################################################
# Function that calculate the average
######################################################################

def List_avg(x):
    List =[]
    for i in range(n):
        List.append(avg(v_N_alpha(x)[i]))
        List.append(avg(v_M_alpha(x)[i]))
        List.append(avg(w_q(x)[i]))
        if i<n-1:
            List.append(avg(w_t(x)[i]))
    return List

######################################################################
# Function that calculate the jump(x) = (x^+ - x^-) n
######################################################################

def List_jump(x):
    List = []
    for i in range(n):
        List.append(jump(totensor(N_alphabeta(x)[i]),n1))
```

```
        List.append(jump(totensor(M_alphabeta(x)[i]),n1))
        List.append(jump(Q_alpha(x)[i],n1))
        if i<n-1:
            List.append(jump(tau_alpha(x)[i], n1))
    return List

######################################################################
# Function used to convert to a vector
######################################################################

def to_vec(x, slice=None):
    if slice is None:
        slice = range(n)
    List = []
    for i in slice:
        List.append(x[4*i][0])
        List.append(x[4*i][1])
        List.append(x[4*i+1][0])
        List.append(x[4*i+1][1])
        List.append(x[4*i+2])
        if i < n-1:
            List.append(x[4*i+3])
    return List


######################################################################
# The Functions : Test, Trial
######################################################################

u = Function(V)
DG = split(u)
u_ = TestFunction(V)
du = TrialFunction(V)
n1 = FacetNormal(mesh)

######################################################################
# The linear, bilinear forms and the boundary conditions for
the Homogeneous laminate
######################################################################
bc = []
bc += [DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 1),
       DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 2),
       DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 3),
```

```
       DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 4)]


a = dot(as_vector(generalized_stress(u_)),
        as_matrix(S_G)*as_vector(generalized_stress(du)))*dx
a += inner(as_vector(EDE_tovec(u_)), as_vector(generalized_displ(du)))*dx
    +inner(as_vector(EDE_tovec(du)), as_vector(generalized_displ(u_)))*dx
a += dot(as_vector(to_vec(List_jump(u_))), as_vector(to_vec(List_avg(du))))*dS
    +dot(as_vector(to_vec(List_jump(du))), as_vector(to_vec(List_avg(u_))))*dS

l = Constant(1)*(dot(as_vector(generalized_displ(u_)), generalized_forces(load))*dx
    +dot(as_vector(generalized_stress(u_)), EP(load))*dx)

######################################################################
# The linear, bilinear forms and the boundary conditions for the
laminate with a circular hole and it's boundary conditions
######################################################################

Neumann_bc = (11, 13, 14)
bc = [DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 12)]

a = dot(as_vector(generalized_stress(u_)), dot(as_tensor(S_G),
        as_vector(generalized_stress(du))))*dx
    + inner(as_vector(EDE_tovec(u_)), as_vector(generalized_displ(du)))*dx
    +inner(as_vector(EDE_tovec(du)), as_vector(generalized_displ(u_)))*dx
    +dot(as_vector(to_vec(List_jump(u_))), as_vector(to_vec(List_avg(du))))*dS
    +dot(as_vector(to_vec(List_jump(du))), as_vector(to_vec(List_avg(u_))))*dS

l = Constant(0)*(dot(as_vector(generalized_displ(u_)), generalized_forces(load))*dx
   + dot(as_vector(generalized_stress(u_)), EP(load))*dx)

for i in range(n):
    a += (dot(dot(totensor(M_alphabeta(u_)[i]), n1), as_vector(v_M_alpha(du)[i]))
         +dot(dot(totensor(M_alphabeta(du)[i]), n1), as_vector(v_M_alpha(u_)[i]))
         +dot(dot(totensor(N_alphabeta(u_)[i]), n1), as_vector(v_N_alpha(du)[i]))
         +dot(dot(totensor(N_alphabeta(du)[i]), n1), as_vector(v_N_alpha(u_)[i]))
         + dot(dot(as_vector(Q_alpha(u_)[i]), n1), w_q(du)[i])
         +dot(dot(as_vector(Q_alpha(du)[i]), n1), w_q(u_)[i]))*ds(Neumann_bc)

layer = 2
l += dot(as_vector(v_N_alpha(u_)[layer]), Constant((1., 0.)))*ds(13)
for j in range(n-1):
    a += (dot(dot(as_vector(tau_alpha(u_)[j]), n1), w_t(du)[j])
```

```
        +dot(dot(as_vector(tau_alpha(du)[j]), n1), w_t(u_)[j]))*ds(Neumann_bc)

#####################################################################
# The linear, bilinear forms and the boundary conditions for Bending
of a laminate with multi-cracking
#####################################################################
Neumann_bc = (2, 4)
bc = [DirichletBC(V[2], Constant((0.,)*(6*n - 1)), 1)]

crack = [1]
Kint = Constant(1e-8)
a = dot(as_vector(generalized_stress(u_)), dot(as_tensor(S_G),
        as_vector(generalized_stress(du))))*dx
    + inner(as_vector(EDE_tovec(u_)), as_vector(generalized_displ(du)))*dx
    +inner(as_vector(EDE_tovec(du)), as_vector(generalized_displ(u_)))*dx
    +dot(as_vector(to_vec(List_jump(u_))), as_vector(to_vec(List_avg(du))))*dS
    +dot(as_vector(to_vec(List_jump(du))), as_vector(to_vec(List_avg(u_))))*dS
    +1/2/Kint*dot(as_vector(to_vec(List_dot(u_), slice=crack))("+"),
             as_vector(to_vec(List_dot(du), slice=crack))("+"))*dS(3)

l = Constant(0)*(dot(as_vector(generalized_displ(u_)), generalized_forces(load))*dx
   + dot(as_vector(generalized_stress(u_)), EP(load))*dx)

for i in range(n):
    a += (dot(dot(totensor(M_alphabeta(u_)[i]), n1), as_vector(v_M_alpha(du)[i]))
         +dot(dot(totensor(M_alphabeta(du)[i]), n1), as_vector(v_M_alpha(u_)[i]))
         +dot(dot(totensor(N_alphabeta(u_)[i]), n1), as_vector(v_N_alpha(du)[i]))
         +dot(dot(totensor(N_alphabeta(du)[i]), n1), as_vector(v_N_alpha(u_)[i]))
         +dot(dot(as_vector(Q_alpha(u_)[i]), n1), w_q(du)[i])
         +dot(dot(as_vector(Q_alpha(du)[i]), n1), w_q(u_)[i]))*ds(Neumann_bc)
    l += dot(w_q(u_)[i], Constant(1.))*ds(2)

for j in range(n-1):
    a += (dot(dot(as_vector(tau_alpha(u_)[j]), n1), w_t(du)[j])
         +dot(dot(as_vector(tau_alpha(du)[j]), n1), w_t(u_)[j]))*ds(Neumann_bc)


#####################################################################
# The problem solver
#####################################################################

ref_problem = LinearVariationalProblem(a, l, u, bcs=bc)
ref_solver = LinearVariationalSolver(ref_problem, solver_parameters=params)
```

```
ref_solver.solve()


######################################################################
# The Total energy
######################################################################
energy =  assemble(0.5*action(action(a,u),u)- action(l,u))
```

Bibliographie

J. Mark Ainsworth and T. Oden. A posteriori error estimation in finite element analysis. 2000.

M.S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, and G.N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3 :[DOI], 2015.

M. S. AlnÃ¦s, A. Logg, K. B. Ãlgaard, M. E. Rognes, and G. N. Wells. Unified form language : A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40(2), 2014.

H. Altenbach. Theories for laminated and sandwich plates. *Mechanics of Composite Materials 34*, pages 243–252, 1998.

SA. Ambartsumian. On theory of bending plates. *Isz Otd Tech Nauk AN SSSR*, 5 :69â77, 1958.

Douglas Arnold, Richard Falk, and Ragnar Winther. Mixed finite element methods for linear elasticity with weakly imposed symmetry. *Mathematics of Computation*, 76(260) : 1699–1723, 2007.

Douglas N Arnold and Franco Brezzi. Mixed and nonconforming finite element methods : implementation, postprocessing and error estimates. *ESAIM : Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 19(1) :7–32, 1985.

Douglas N Arnold and Ragnar Winther. Mixed finite elements for elasticity. *Numerische Mathematik*, 92(3) :401–419, 2002.

R.C. Averill. Static and dynamic response of moderately thick laminated beams with damage. *Composites Engineering*, 4 :381–395, 1994.

M. Aydogdu. Comparison of various shear deformation theories for bending, buckling, and vibration of rectangular symmetric cross-ply plate with simply supported edges. *Journal of Composite Materials*, 2006.

B. S. Aziz. The mathematical foundation of the finite element method with applications to partial differential equations. *Academic Press, New York*, 33 :1975â1996, 1972.

I. Babuska and W. C. Rheinboldt. Aposteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10) :1597â1615, 1987.

E.J. Barbero and J.N. Reddy J.N. Modeling of delamination in composite laminates using a layerwise plate theory. *International Journal of Solids and Structures*, 28 :373–388, 1991.

R. Baroud. Development and implementation of numerical models for the study of multilayered plates. materials. *UniversitÃ© Paris-Est ; Universite Saint-Joseph (Beyrouth). Ecole superieure dâingenieurs de Beyrouth, 2016. English.*, 2016.

R. Baroud, K. Sab, J.F. Caron, and F. Kaddah. A statically compatible layerwise stress model for the analysis of multilayered plates. *International Journal of Solids and Structures*, 69 :11–24, 2016.

KJ. Bathe and F. Brezzi. A simplified analysis of two plate bending elements-the mitc4 and mitc9 elements. *Proceedings, Numerical Methods in Engineering : Theory and Applications,Swansea, UK : University College*, 21, 1987.

KJ Bathe and E. Dvorkin. A formulation of general shell elements - the use of mixed interpolation of tensorial components. *International Journal for Numerical Methods in Engineering*, 22 :697–722, 1986.

KJ. Bathe, F. Brezzi, and SW. Cho. The mict7 and mitc9 plate bending elements. *Computers Structures*, 32(3-4) :797–814, 1989.

I. Benedetti and A. Milazzo. Advanced models for smart multilayered plates based on reissner mixed variational theorem. *Composites Part B*, 119 :215–229, 2017.

Jeremy Bleyer. Numerical tours of computational mechanics with FEniCS. 2018. doi : 10.5281/zenodo.1287832.

Franco Brezzi and Michel Fortin. *Mixed and hybrid finite element methods*, volume 15. Springer Science & Business Media, 2012.

ML. Bucalem and E. Dvorkin. Higher-order mitc general shell elements. *International Journal for Numerical Methods in Engineering*, 36(37) :29–54, 1993.

J. Caron and A. Ehrlacher. Modelling the kinetics of transverse cracking in composite laminates. *Composites Science and Technology*, 57 :1261–1270, 1997.

J. F. Caron. Modélisation de la cinétique de fissuration transverse en fatigue dans les stratifés. *Thèse de l'Ecole Nationale des Ponts et Chaussées*, 1993.

J.F. Caron, A. Diaz Diaz, R.P. Carreria, A. Chabot, and A. Ehrlacher. Multi-particle modelling for the prediction of delamination in multi-layered materials. *Composites Science and Technology*, 66 :755–765, 2006.

R. P. Carreira. Validations par éléments finis des modèles multiparticulaires des matériaux multicouches m4. *Thèse de Doctorat de lâENPC*, 1998.

E. Carrera. A class of two-dimensional theories for anisotropic multilayered plates analysis. *Accademia delle Scienze di Torino, Memorie Scienze Fisiche*, 19(20) :1–39, 1995.

E. Carrera. Mc z0 requirements-models for the two dimensional analysis of multilayered structures. *Composite Structures*, 37(37) :73–84, 1997.

E. Carrera. Mixed layer-wise models for multilayered plates analysis. *Composite Structures 43*, 43 :57–70, 1998.

E. Carrera. Developments in the modeling of multilayered plates and shells based upon reissnerâs mixed equation. *In : XIV congresso AIMETA, Como 6-9*, 1999.

E. Carrera. Multilayered shell theories that account for a layer-wise mixed description. part i. governing equations. *American Institute of Aeronautics and Astronautics Journal*, 37 :1107–1116, 1999a.

E. Carrera. Multilayered shell theories that account for a layer-wise mixed description. part ii. numerical evaluations. *American Institute of Aeronautics and Astronautics Journal*, 37 :1117–1124, 1999b.

E. Carrera. An assessment of mixed and classical theories on global and local response of multilayered orthotropic plates. *Composite Structures*, 50 :183–198, 2000.

E. Carrera. Theories and finite elements for multilayered , anisotropic, composite plates and shells. *Archives of computational Methods in Engineering*, 9 :87–140, 2002.

E. Carrera. Theories and finite elements for multilayered plates and shells : a unified compact formulation with numerical assessment and benchmarking. *Archives of Computational Methods in Engineering*, 10 :15–96, 2003.

E. Carrera. On the use of the murakami's zig-zag function in the modeling of layered plates and shells. *Computers Structures*, 82 :541–554, 2004.

E. Carrera and L. Demasi. Multilayered finite plate element based on reissner mixed variational theorem. part ii : numerical analysis. *International Journal for Numerical Methods in Engineering*, 55(2) :53–96, 2002a.

E. Carrera and L. Demasi. Sandwich plates analyses by finite element method and reissnerâs mixed theorem. *Sandwich V, Zurich, k*, 5(7) :301–313, 2002b.

E. Carrera and L. Demasi. Multilayered finite plate element based on reissner mixed variational theorem. part i : theory. *International Journal for Numerical Methods in Engineering*, 55 :191–231, 2002c.

E. Carrera and E. Zappino. Analysis of complex structures coupling variable kinematics one-dimensional models. *ASME 2014 International Mechanical Engineering Congress and Exposition, Montreal, Quebec, Canada*, 2014.

E. Carrera, A. Pagani, and S. Valvano. Multilayered plate elements accounting for refined theories and node-dependent kinematics. *Composites Part B*, 114 :189–210, 2017.

R.P. Carreria, J.F. Caron, and A. Diaz Diaz. Model of multilayered materials for interface stresses estimation and validation by finite element calculations. *Mechanics of Materials*, 34 :217–230, 2002.

A. Cecchi and K. Sab. A homogenized reissner-mindlin model for orthotropic periodic plates : Application to brickwork panels. *International Journal of Solids and Structures*, 44 :6055–6079, 2007.

A. Chabot. Analyse des efforts Ã lâinterface entre les couches des matériaux composites à lâaide de modèles multiparticulaires de matériaux multicouches (m4). *Thèse. Ecole Nationale des Ponts et Chaussées*, 1997.

A. Chabot and P. Deep. 2d multilayer analysis of electrified roads with charging box discontinuities. *Paper presented at the 13th ISAP Conference on Asphalt Pavements, Fortaleza, CearÃ¡,Brazil*, 2018.

A. Chabot and P. Deep. 2d multilayer solution for an electrified road with a built-in charging box. *Road Materials and Pavement Design*, 20 :sup2 :590–603, 2019.

A. Chabot and A. Ehrlacher. Méthode de construction de modèles multiparticulaires des matériaux multicouches (modèles m4) pour lâanalyse des efforts à lâinterface entre les couches de plaques composites. *Rapport de recherche du CERAM (ENPC)*, 1998a.

A. Chabot, Q. D. Tran, A. Ehrlacher, and P. Tamagny. Modèle simplifié pour le calcul de chaussées. *16ème Congrès FranÃ§ais de Mécanique, 1-3 Septembre, Nice*, 2003.

A. Chabot, P. Tamagny, Q.D. Tran, and A. Ehrlacher. Modeling of stresses for cracking in pavements. *5th International CROW-workshop, 31 mars-2 avril, Istanbul, Turkey*, 2004a.

A. Chabot, Q. D. Tran, and B. Pouteau. Simplified modelling of a cracked composite pavement. *First International Elsevier Conference on Failure Analysis, Lisbonne 12-14 juillet*, 2004b.

A. Chabot, Q. D. Tran, and A. Ehrlacher. A simplified modeling for cracked pavements - modèle simplifié pour le calcul des chaussées. *Bulletin des Laboratoires des Ponts et chaussées (http ://www.lcpc.fr/en/sources/blpc/index.php)*, 12 :105–120, 2005.

A. Chabot, Q. D. Tran, and A. Ehrlacher. A modeling to understand where a vertical crack can propagate in pavements. *International Conference on Advanced Characterization of Pavement and Soil Engineering Materials*, pages 431–440, 2007.

A. Chabot, M. Hun, and F. Hammoum. Mechanical analysis of a mixed mode debonding test for ââcompositeââ pavements. *Construction and Building Materials*, 40 :1076–1088, 2013.

C. Chinosi, M. Cinefra, L. Della Croce, and E. Carrera. Reissnerâs mixed variational theorem toward mitc finite elements for multilayered plates. *Composite Structures*, 99 (4) :43–52, 2013.

C. Chinosi, L. Della Croce, M. Cinefra, and E. Carrera. Approximation of anisotropic multilayered plates through rmvt and mitc elements. *Composite Structures*, 158 :252–261, 2016.

M. Cho and R.R. Parmerter. Efficient higher order composite plate theory for general lamination configurations. *American Institute of Aeronautics and Astronautics Journal*, 31 :1299–1305, 1993.

C.H. Chue and C.I. Liu. Disappearance of free-edge stress singularity in composite laminates. *Composite Structures*, 56 :111–129, 2002.

P. Ciarlet and P. Destuynder. Justification of the 2-dimensional linear plate mode. *Journal de mecanique*, 18 :315–344, 1979.

P. G. Ciarlet. The finite element method for elliptic problems. *Access Online via Elsevier*, 1978.

Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2) :1319–1365, 2009.

F. W. Crossman and A. S. D. Wang. The dependence of transverse craking and delamination on ply thickness in graphite/epoxy laminates. *Damage in Composite Materials, ASTM ATP 775, K. L. Reifsnider, Ed, American Society for Testing and Materials*, pages 118–139, 1982.

F. Daghia, S. de Miranda, F. Ubertini, and E. Viola. A hybrid stress approach for laminated composite plates within the first-order shear deformation theory. *International Journal of Solids and Structures*, 45 :1766–1787, 2008.

J. Dallot and K. Sab. Limit analysis of multi-layered plates. Part II :shear effects. *Journal of the Mechanics and Physics of Solids*, 56 :561–580, 2008.

P. Deep. Mechanical analysis of an eroad discontinuities. *(Unpublished masterâs thesis). Ecole Centrale de Nantes (ECN), France*, 2017.

G. Dhatt and G. Touzot. Une présentation de la méthode des éléments finis. *Maloine, Paris*, 1984.

M. Di Sciuva. A refined transverse shear deformation theory for multilayered anisotropic plates. *ATTI DELLA ACCADEMIA DELLE SCIENZE DI TORINO*, 118 :279–295, 1984.

M. Di Sciuva. Bending, vibration and buckling of simply supported thick multilayered orthotropic plates : An evaluation of a new displacement model. *Journal of Sound and Vibration*, 105 :425–442, 1986.

A. Diaz Diaz, J.F. Caron, and R.P. Carreira. Software application for evaluating interfacial stresses in inelastic symmetrical laminates with free edges. *Composite Structures*, 58 : 195–208, 2002.

A. Diaz Diaz, J. F. Caron, and A. Ehrlacher. Analytical determination of the modes i, ii and iii energy release rates in a delaminated laminate and validation of a delamination criterion. 2007.

A. Diaz Diaz. Delaminage des materiaux multicouches : Phenomenes, modeles et criteres. *Theses. Ecole Nationale des Ponts et Chaussées.*, 2001.

J. Dundurs. Edge-bonded dissimilar orthogonal elastic wedges under normal and shear loading. transactions of the asme. *Journal of Applied Mechanics*, pages 650–652, 1969.

Baudouin Fraeijs de Veubeke. Displacement and equilibrium models in the finite element method. *Stress analysis*, pages chapter–9, 1965.

P. J. Frey and P. L. George. Mesh generation : Application to finite elements, second edition. 2008.

K. W. Garett and J. E. Bailley. Multiple transverse fracture in 90Â° cross-ply laminates of a glass fibre-reinforced polyester. *Journal of Materials Science*, 12 :157–168, 1977.

P. Gaudenzi, R. Barboni, and A. Mannini. A finite element evaluation of single-layer and multi-layer theories for the analysis of laminated plates. *Computers Structures.*, 30 : 427–440, 1995.

Thomas H Gibson, Lawrence Mitchell, David A Ham, and Colin J Cotter. Slate : extending firedrake's domain-specific abstraction to hybridized solvers for geoscience and beyond. *Geoscientific model development*, 13(2) :735–761, 2020.

Shihua Gong, Shuonan Wu, and Jinchao Xu. New hybridized mixed methods for linear elasticity and optimal multilevel solvers. *Numerische Mathematik*, 141(2) :569–604, 2019.

V. Gulizzi, I. Benedetti, and A. Milazzo. A high-resolution layer-wise discontinuous galerkin formulation for multilayered composite plates. *Composite Structures*, 242 :112–137, 2020.

Murakami H. Laminated composite plate theory with improved in-plane responses. *ASME Journal of Applied Mechanics*, pages 53–66, 1986.

R. Hadj-Ahmed, G. Foret, and A. Ehrlacher. Stress analysis in adhesive joints with a multiparticle model of multilayered materials (m4). *International Journal of Adhesion and Adhesives*, 21 :297–307, 2001.

L.H. He. A linear theory of laminated shells accounting for continuity of displacements and transverse shear stresses at layer interfaces. *International Journal of Solids and Structures*, 31 :613–627, 1994.

M. Hun. Influence de lâeau dans le dÃ©collement par flexion de bicouches de chaussée urbaine. *Thèse de Doctorat de lâED SPIGA, ECN.*, 2012.

M. Hun, A. Chabot, and F. Hammoum. Analyses mécaniques dâune structure bicouches délaminantes par flexion 4 points. *20ème Congrès FranÃ§ais de Mécanique (CFM2011), BesanÃ§on, France*, 2011.

U. Incardi. Higher-order zig-zag model for analysis of thick composite beams with inclusion of transverse normal stress and sublaminates approximations. *Composites Part B : Engineering*, 32 :343–354, 2001.

M. Karama, B.A. Harb, S. Mistou, and S. Caperaa. Bending, buckling and free vibration of laminated composite with a transverse shear stress continuity model. *Composites Part B : Engineering*, 29 :223–234, 1998.

R. Kienzler. On consistent plate theories. *Archive of Applied Mechanics*, 72 :229–247, 2002.

R. C. Kirby. Fiat, a new paradigm for computing finite element basis functions. *ACM Transactions on Mathematical Software*, 30(4) :502–516, 2004.

R. C. Kirby and A. Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(3) :417–444, 2006.

Robert C Kirby and Lawrence Mitchell. Solver composition across the pde/linear algebra barrier. *SIAM Journal on Scientific Computing*, 40(1) :C76–C98, 2018.

Andreas Klöckner. Loo.py : transformation-based code generation for GPUs and CPUs. In *Proceedings of ARRAY '14 : ACM SIGPLAN Workshop on Libraries, Languages, and Compilers for Array Programming*, Edinburgh, Scotland., 2014. Association for Computing Machinery. doi : {10.1145/2627373.2627387}.

P. Ladevéze. Comparaison de modeles de milieux continus. *Thése de doctorat.*, 1975.

A. Lebée and K. Sab. A Bending-Gradient model for thick plates. Part I : theory. *International Journal of Solids and Structures*, 48 :2878–2888, 2011a.

A. Lebée and K. Sab. A Bending-Gradient model for thick plates. Part II :closed form solutions for cylindrical bending of laminates. *International Journal of Solids and Structures*, 48 :2889–2901, 2011b.

A. Lebée and K. Sab. Homogenization of thick periodic plates : application of the Bending-Gradient plate theory to a folded core sandwich panel. *International Journal of Solids and Structures*, 49 :2778–2792, 2012.

D. Leguillon. A method based on singularity theory to predict edge delamination of laminates. *International Journal of Fracture*, 100 :105–120, 1999.

D. Leguillon and E.Sanchez-Palencia. Computation of singular solutions in elliptic problems and elasticity. *WILEY-VCH Verlag.*, 68, 1987.

A. Lerpiniere, J.F. Caron, A. Diaz Diazand, and K. Sab. The ls1 model for delamination propagation in multilayered materials at $0^0/\theta^0$ interfaces : a comparison between experimental and finite elements strain energy release rates. *International Journal of Solids and Structures* ., 51 :3973–3986, 2014.

M. Levinson. An accurate, simple theory of the statics and dynamics of elastic plates. *Mechanics Research Communications*, 7 :343–350, 1980.

M. Levy. Comptes rendues. 129 :535â539, 1899.

Y. Liu. A refined shear deformation plate theory. *Int. J. for Computational Methods in Engineering Science and Mechanics*, 12 :141–149, 2011.

A. Logg and G. N. Wells. Dolfin : Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2), 2010.

A. Logg, K.-A. Mardal, and G. N. Wells. Automated solution of partial differential equations by the finite element method. *Springer*, 2012a.

A. Logg, K.A. Mardal, and G.N. Wells et al. Automated solution of differential equations by the finite element method. *Springer*, 2012b.

D.S. Marco and I. Ugo. Discrete-layer models for multilayered shells accounting for interlayer continuity. *Meccanica*, 28 :281–291, 1993.

R.D. Mindlin. Influence of rotary inertia and shear on flexural motions of isotropic, elastic plates. *Journal of Applied Mechanics*, 18 :31–38, 1951.

C. Mittelsteda and W. Becker. Asymptotic analysis of stress singularities in composite laminates by the boundary finite element method. *Composite Structures*, 71 :210–219, 2005.

F. Moleiro, C.M. Mota Soares, C.A. Mota Soares, and J.N. Reddy. Layerwise mixed leastsquares finite element models for static and free vibration analysis of multilayered composite plates. *Computers Structures*, 92 :28–38, 2010.

F. Moleiro, C.M. Mota Soares, C.A. Mota Soares, and J.N. Reddy. A layerwise mixed leastsquares finite element model for static analysis of multilayered composite plates. *Computers Structures*, 89 :30–42, 2011.

F. Moleiro, C.M. Mota Soares, C.A. Mota Soares, and J.N. Reddy. Assessment of a layerwise mixed least-squares model for analysis of multilayered piezoelectric composite plates. *Computers Structures*, 108-109 :14–30, 2012.

F. Moleiro, C.M. Mota Soares, C.A. Mota Soares, and J.N. Reddy. Layerwise mixed models for analysis of multilayered piezoelectric composite plates using least-squares formulation. *Computers Structures*, 119(1) :34–49, 2015.

F. Moleiro, V.M. Franco Correi, A.L. AraÃºjo, C.M. Mota Soares, A.J.M. Ferreira, and J.N. Reddy. Deformations and stresses of multilayered plates with embedded functionally graded material layers using a layerwise mixed model. *Composites Part B*, 156 : 274–291, 2019.

F. Moleiro, E. Carrera, E. Zappino, G. Li, and M. Cinefra. Layerwise mixed elements with node-dependent kinematics for globalâlocal stress analysis of multilayered plates using high-order legendre expansions. *Computer Methods in Applied Mechanics and Engineering*, 359(11) :27–64, 2020.

C.M. Dakshina Moorthya and J.N. Reddy. Modeling of delamination using a layerwise element with enhanced strains. *International Journal for Numerical Methods in Engineering*, 43 :755–779, 1998.

H. Murakami. Laminated composite plate theory with improved in-plane responses. *ASME Proc PVP Conf, New Orleans*, 98 :57–63, 1985.

M. Murthy. An improved transverse shear deformation theory for laminated anisotropic plate. *Technical Report.NASA*, 1981.

H. Nasser and A. Chabot. Two-dimensional software for analysing mechanical fields in elastic cracked pavements. *Proceedings of the Fifteenth International Conference on Civil, Structural and Environmental Engineering Computing Civil-Comp Press, Stirlingshire, UK, Paper 214*, 2015.

H. Nasser and A. Chabot. A half-analytical elastic solution for 2d analysis of cracked pavements. *Advances in Engineering Software*, 117 :107–122, 2018.

H. Nasser, J. M. Piau, O. Chupin, and A. Chabot. M4-5n numerical solution using the mixed fem,validation against the finite difference method. *8th RILEM International Conference on Mechanisms of Cracking and Debonding in Pavements.*, 13 :363–369, 2016.

H. Nasser, O. Chupin, J. M. Piau, and A. Chabot. Mixed fem for solving a plate type model intended for analysis of pavements with discontinuities. *Road Materials and Pavement Design, Taylor Francis*, 3 (19) :496–510, 2018a.

Hanan Nasser, Olivier Chupin, J-M Piau, and Armelle Chabot. Mixed fem for solving a plate type model intended for analysis of pavements with discontinuities. *Road Materials and Pavement Design*, 19(3) :496–510, 2018b.

T.K. Nguyen, K. Sab, and G. Bonnet. First-order shear deformation plate models for functionally graded materials. *Composite Structures 83*, pages 25–36, 2008.

V.T. Nguyen and J.F. Caron. A new finite element for free edges effect analysis in laminated composites. *Composite Structures 84*, pages 1538–1546, 2006.

C.V. Nielsen, W. Zhang L.M. Alves, N. Bay, and P.A.F. Martins. Meshing and remeshing. in : Modeling of thermo-electro-mechanical manufacturing processes. *SpringerBriefs in Applied Sciences and Technology. Springer, London. https ://doi-org.extranet.enpc.fr/10.1007/978-1-4471-4643-8*$_5$, 2013.

N. Pagano. Stress fields in composite laminates. *International Journal of Solids and Structures*, 14 :385–400, 1978.

N. Pagano and R. Pipes. Composite laminates under uniform axial extension. *Journal of Composite Materials*, pages 538–548, 1970.

N. Pagano and S. Soni. Global-local laminate variational model. *International Journal of Solids and Structures*, 19 :207–228, 1983.

Florian Rathgeber, David A Ham, Lawrence Mitchell, Michael Lange, Fabio Luporini, Andrew TT McRae, Gheorghe-Teodor Bercea, Graham R Markall, and Paul HJ Kelly. Firedrake : automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3) :1–27, 2016.

J. N. Reddy. A mechanics of laminated composite plates and shells - theory and analysis. *CRC Press, Boca Raton, Florida, 2nd Edition*, 2004.

J.N. Reddy. A simple higher-order theory for laminated composite plates. *Journal of Applied Mechanics*, 51 :745–752, 1984.

E. Reissner. A consistent treatment of transverse shear deformations in laminated anisotropic plates. *AIAA Journal*, 10 :716–718, 1972.

E. Reissner. On a certain mixed variational theorem and a proposed application. *International Journal for Numerical Methods in Engineering*, 20(7), 1984.

E. Reissner and Y. Stavsky. Bending and stretching of certain types of heterogeneous aeolotropic elastic plates. *Journal of Applied Mechanics*, 28 :402–408, 1961.

D.H. Robbins and J.N. Reddy. Modeling of thick composites using a layerwise laminate theory. *International Journal for Numerical Methods in Engineering*, 36 :655–677, 1993.

N. Saeedi, K. Sab, and J.F. Caron. Delaminated multilayered plates under uniaxial extension. Part I : analytical analysis using a layerwise stress approach. *International Journal of Solids and Structures*, 49 :3711–3726, 2012a.

N. Saeedi, K. Sab, and J.F. Caron. Delaminated multilayered plates under uniaxial extension. Part II : very efficient layerwise mesh strategy for the prediction of delamination onset. *International Journal of Solids and Structures*, 49 :3727–3740, 2012b.

N. Saeedi, K. Sab, and J.F. Caron. Cylindrical bending of multilayered plates with multidelamination via a layerwise stress approach. *Composite Structures*, 95 :728–739, 2013a.

N. Saeedi, K. Sab, and J.F. Caron. Stress analysis of long multilayered plates subjected to invariant loading : analytical solutions by a layerwise stress model. *Composite Structures*, 100 :307–322, 2013b.

L. Salha, J. Bleyer, K. Sab, and J. Bodgi. Mesh-adapted stress analysis of multilayered plates using a layerwise model. *Advanced Modeling and Simulation in Engineering Sciences*, 7, 2020.

S. A. Sauter and C. Schwab. A posteriori error estimation. in : Boundary element methods. *Springer series in computational mathematics. Springer, London.*, 2011.

S.Botello, E. Onate, and J.M. Canet. A layer-wise trinagle for analysis of laminated composite plates and shells. *Computers Structures.*, 70 :635–646, 1999.

G. Shi. A new simple third-order shear deformation theory of plates. *International Journal of Solids Structures*, 44 :4299–4417, 2007.

K. P. Soldatos. A transverse shear deformation theory for homogeneous monoclinic plates. *Acta Mechanica*, 94(3â4) :19 :95â220, 1992.

Tianjiao Sun, Lawrence Mitchell, Kaushik Kulkarni, Andreas Klöckner, David A Ham, and Paul HJ Kelly. A study of vectorization for matrix-free finite element methods. *arXiv preprint arXiv :1903.08243*, 2019.

K. Swaminathan and D. Ragounadin. Analytical solutions using a higher-order refined theory for the static analysis of antisymmetric angle-ply composite and sandwich plates. *Composite Structures*, 64 :405–417, 2004.

V. L. Tahari, C. Henaff-Gardin, and Lafarie-Frenot. Apparition et propagation des endommagements par fatigue dans les stratifiés carbone/epoxyde (0°/45°). *Comptes-Rendus des JNC9*, 1 :22–24, 1994.

K. Tarun, D. R. J. Owen, and O. C. Zienkiewicz. A refined higher-order c0 plate bending element. *Computers and Structures*, 15 :177–183, 1982.

N.D. Thai, M. DâOttavio, and J. F. Caron. Bending analysis of laminated and sandwich plates using a layer-wise stress model. *Composite Structures*, 96 :1 :35–42, 2013.

T.C.T. Ting and S.C. Chou. Edge singularities in anisotropic composites. *International Journal of Solids and Structures*, 17 :1057–1068, 1981.

A. Toledano and H. Murakami. A high-order laminated plate theory with improved in-plane responses. *International Journal of Solids and Structures*, 23(1) :11–31, 1987a.

A. Toledano and H. Murakami. A composite plate theory for arbitrary laminate configurations. *ASME J appl Mech*, 54(18) :1–9, 1987b.

M. Touratier. An efficient standard plate theory. international. *Journal of Engineering Science*, 29 :901–916, 1991.

M. Touratier. A refined theory of laminated shallow shells. *International Journal of Solids and Structures*, 29 :1401–1415, 1992.

Q. D. Tran, A. Chabot, A. Ehrlacher, and P. Tamagny. A simplified modelling for cracking in pavements. *5th International RILEM Conference Cracking in Pavements,*, 2004.

Ya. S. Uflyand. Wave propagation by transverse vibrations of beams and plates (in russian). *Journal of Applied Mathematics and Mechanics*, 12 :287–300, 1948.

I. Vekua. Shell theory : general methods of construction. monographes, advanced texts and surveys in pure and applied mathematics. *John Wiley and sons, New York,*, 1985.

P Vidal and O Polit. A family of sinus finite elements for the analysis of rectangular laminated beams. *Composite Structures*, 84(1) :56–72, 2008.

P Vidal and O Polit. A sine finite element using a zig-zag function for the analysis of laminated composite beams. *Composites Part B : Engineering*, 42(6) :1671–1682, 2011.

P. Vidal, L. Gallimard, and O. Polit. Proper generalized decomposition and layer-wise approach for the modeling of composite plate structures. *International Journal of Solids and Structures*, 50(14-15) :2239–2250, 2013.

C. M. Wang, J. N. Reddy, and K. H. Lee. Shear deformable beams and plates : Relationships with classical solutions. *Elsevier, Oxford, UK.*, 2000.

S.S. Wang and I. Choi. Boundary-layer effects in composite laminates. Part I : free-edge stress singularities. *Journal of Applied Mechanics*, 49 :541–548, 1982.

C.T. Whitney, J.M.and Sun. A higher-order theory for extensional motion of laminates composites. *Journal of Sound and Vibration*, 30 :85–97, 1973.

J.N. Whitney and N.J. Pagano. Shear deformation in heterogeneous anisotropic plates. *Journal of Applied Mechanics*, 37 :1031–1036, 1970.

P. Yang, C.H. Norris, and Y. Stavsky. Elastic wave propagation in heterogeneous plates. *International Journal of Solids and Structures*, 2 :665–684, 1996.

E. Zappino, G. Li, A. Pagani, E. Carrera, and A.G. de Miguel. Use of higher-order legendre polynomials for multilayered plate elements with node-dependent kinematics. *Composite Structures*, 202 :222–232, 2018.

Y.X. Zhang and C.H. Yang. Recent developments in finite element analysis for laminated composite plates. *Composite Structures*, 88 :147–157, 2009.

O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineerng analysis. *International Journal for Numerical Methods in Engineering*, 24(2) :337â357, 1987.