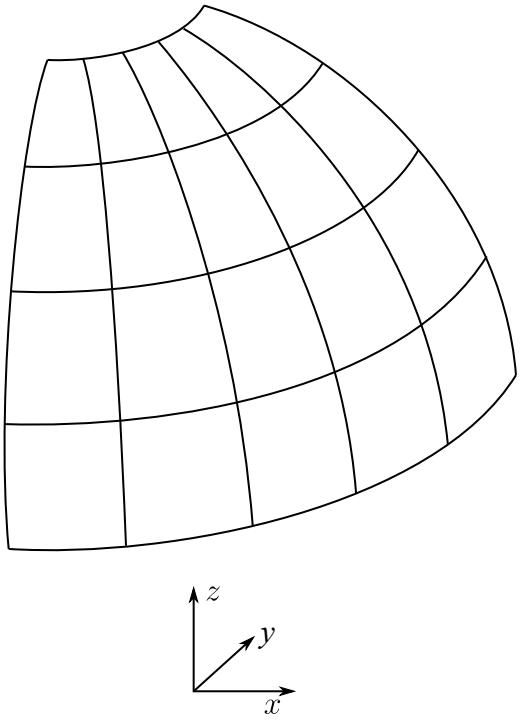
An enhanced 4-node shell element for laminated composites



Design of Mechanical Systems
Master's Thesis
Pau Capella and Guillem Gall
Aalborg University
June 1st 2018

AALBORG UNIVERSITY
STUDENT REPORT

Department of Materials
and Production
Fibigerstraede 16, 9220 Aalborg Øst

Title:

An enhanced 4-node shell element for laminated composites.

Project:

Master's thesis

Period:

01/02/18 - 01/06/18

Group:

14.23A

Members:

Pau Capellá
Guillem Gall Trabal

Supervisor:

Erik Lund

No. pages in the report: 81
Submission: 01-06-2018

Synopsis:

Computationally fast quadrilateral elements that are capable of modeling laminated shell structures are of great attraction for optimization purposes. In this thesis, a 4-node linear and non-linear shell element is formulated and partially implemented in the MUST platform.
Because of the enhancements chosen to avoid the shear locking effect, the covariant base and the kinematics of curved geometries are introduced. Later on, the variational formulation for the 4-node shell element is derived and discretized by means of finite elements. The details of the numerical implementation of the element are also provided.
Finally, the buckling effect and the linear buckling formulation for the element are described and implemented in Matlab and a series of numerical tests are performed to assess the behavior of the element.

Preface

This Master's thesis is carried out by two students of the MSc. in Design of Mechanical Systems (DMS) at Aalborg University. It is the result of a job accomplished during the period from February 2018 to June 2018, and takes as a starting point the results obtained from a DMS 3^{rd} semester project done by the same pupils.

This work is aimed for an audience that shows a great interest in the Finite Elements Method and its formulation, particularly when needing deep understanding on how to model laminated shell structures in FEM.

The following software has been used:

- Microsoft Visual Studio Enterprise 2015, Version 14.0.2543, Update 3
- Matlab Release 2016a and 2016b
- ANSYS Mechanical, Release 18.2 and Release 19.0
- Inkscape, 0.17 version
- MikTex, Release 2.9
- Texmaker, Release 4.5

Included figures and tables are numbered with respect to the chapter number. Figures and tables are distinguished implying that, for instance, both figure 1.1 and table 1.1 may exist.

A list of the cited references is to be found at the end of the report. Reference list entries contain all the information needed to follow up the source. As the referencing style is Chicago, the entries are arranged alphabetically by the primary author's surname. Likewise, the used method for citing within the text is Chicago citation style, which is an "author-date" system, so the citation in the text consists of the author(s) name and year of publication given wholly or partly in round brackets.

The Master's thesis has been supervised by professor Ph.D. Erik Lund to whom we thank for all the help and great leading of such a challenging project.

Aalborg University, June 1st, 2018

Pau Capellá Pujol

Guillem Gall Trabal

Abstract

Laminated composite shell structures are becoming increasingly popular because of their mechanical properties versus weight beneficial relation. The aim of this study is to develop a robust finite shell element formulation for the analysis of laminated composites. The 4-node shell element coded in this work is desired to solve for linear static analysis, linear buckling and geometrically non-linear analysis.

The element routine is implemented in the MUltidisciplinary Synthesis Tool (MUST), under development at the Department of Materials and Production at Aalborg University since 1998. MUST is a system used for design, analysis and optimization for structural, fluid and thermal problems coded in Fortran90. Therefore, the 4-node shell element of this thesis is implemented in Fortran90 and fitted adequately in the system.

The starting point of this project is a 4-node flat shell element enhanced with the EAS (Simo and Rifai, 1990) and MITC (Dvorkin and Bathe, 1984) methods. It was coded in Matlab as a 3rd semester project in the DMS master's program. The aforementioned element can solve linear static problems and it is not generalized to the 3D space. Thus, a generalized, non-linear enhanced formulation is needed.

The thesis derives the continuum based theories needed to model shell elements and laminated structures. From there, a derivation of the continuum mechanics in curvilinear is done so that the reader gathers all the needed tools to understand the three-field variational derivation (Washizu, 1975) of the element's governing equation. Once the governing equations are obtained, they are discretized to formulate the finite elements solution.

Finally, the discrete equations to be solved are obtained and they describe a 4-node finite element capable of modeling multi-layered structures. It is enhanced with the EAS and MITC methods to tackle, in this way, the in-plane and out-of-plane locking the isoparametric element displays before any improvement implementation.

The 4-node shell laminated element passes all patch tests for both membrane and bending loading situations. Good results are obtained for laminated and non-laminated elements in both linear and linear bucking tests, regardless the geometry and orientation of the structure. In the case of geometrically non-linear situations the element performs well when bending is not involved and the structure's geometry is not curved. Therefore, a bug in the code appears to be in the geometrically non-linear (GNL) implementation regarding the rotation degrees of freedom or a mistake in transforming the mesh to the local element base. Further work is to be done on debugging the GNL implementation of the element.

Table of contents

Nomenclature

Variables

$[k_0]$	Linear element stiffness matrix
$[k_T]$	Tangent element stiffness matrix
$[k_\sigma] \equiv [G_{IJ}]$	Element stress stiffness matrix
$[k_L]$	Initial displacement stiffness matrix
$[H_e]$	Incompatible element stiffness matrix
$[L_e]$	Element stiffness matrix relating incompatible and compatible DOF
$\{R\}$	Global residual of forces
$\{r\}$	Element residual of forces
h	Element thickness
N	Shape functions
$\{\Phi\}$	Vector of surface tractions
$\{b\}$	Vector of body forces
$\{F_{ext}\}$	Global external force vector
$\{f_{ext}\}$	Element external force vector
$\{f_{int}\}$	Element internal force vector
$\{D\}$	Global displacement vector
$\{d\}$	Element displacement vector
α	Rotational d.o.f in the r direction
β	Rotational d.o.f in the s direction
$[B_0]$	Linear strain-displacement matrix
$[B_{NL}]$	Non-linear strain-displacement matrix
$[B_I]$	Linear + Non-linear strain-displacement matrix
$[C]$	Constitutive matrix
$[\bar{C}]$	Constitutive matrix in fiber orthonormal base
$[G]$	Matrix of covariant components
$[H]$	Derivatives coupling matrix
$[N]$	Interpolation matrix
$[Q]$	Shape function derivatives matrix
$[J]$	Jacobian matrix
$\|\ \|$	Determinant
$[\Gamma]$	Inverse of Jacobian matrix
ε_{ij}	Green-Lagrange strain tensor
$\{\varepsilon_{ij}\}$	Strain tensor as vector
$[\varepsilon_{ij}]$	Strain tensor as matrix
η_{ij}	Non-linear part of Green-Lagrange strain tensor
σ	Cauchy stress
S	Second Piola-Kirchoff stress
$\{u\}$	Displacement vector
$\{\alpha\}$	Incompatible DOF vector
g_{ij}	Metric tensor
$[M]$	Incompatible modes interpolation matrix

$\{X\}$	Position vector of a point in the reference system
$\{x\}$	Position vector of a point in the current state
Π	Total potential energy
Π_{ext}	Potential done by the external forces
Π_{int}	Potential of the internal forces
$[T]$	Transformation matrix (Cook et al., 2001)
U	Strain energy
W_0	Initial stored energy
$[\Lambda]$	3D rotation matrix
$x_i \equiv r, s, t$	Element natural convective coordinates
$\{g_i\}$	Covariants in the reference system
$\{G_i\}$	Covariants in the current deformed state
$\{g^i\}$	Contravariants in the reference system
$\{G^i\}$	Contravariants in the current deformed state

Coordinate systems

(x, y, z)	Global Cartesian coordinate system
(r, s, t)	Natural convective element coordinate system or curved coordinate system
$(\{e_1\}, \{e_2\}, \{e_3\})$	Global Cartesian coordinate system
$(\{d_1\}, \{d_2\}, \{d_3\})$	Local Cartesian coordinate system
$(\{g_1\}, \{g_2\}, \{g_3\})$	Covariant base coordinate system at a desired point

Subscripts and Superscripts

$\tilde{}$	Referring to a quantity calculated in the local covariant base
$_{app}$	Referring to an approximated quantity or field
$_{e}$	Refers to a field or quantity element-wise
$_{I}$, ^{I}	Referring to a node
^{2D}	Referring to the plane isoparametric shape function
$\hat{}$	Refers to incompatible enhancing fields
^{c}	Refers to compatible fields
$_{layer}$	Refers to any quantity layer-wise
$_{1,2,3}$	Refers to any strain/stress component in a FEM context
$_{x,y,z}$	Refers to any strain/stress component in Continuum Mechanics context
$_{r,s,t}$	Refers to any strain/stress component in element natural mapping

1 Introduction

The work done in this Master's thesis belongs to the general category of Computational Methods in Engineering. In particular, numerical methods for designing or analyzing within the field of structural mechanics. The general aim of this study is to develop a finite element that serves as a basis for future work on structural optimization problems involving laminated composite structures for linear static analysis, linear buckling analysis and geometrically nonlinear analysis. In order to do so, a robust, stable finite element formulation is required, which is the object of this study.

Composite materials trend in the industry

Composite materials are utilized extensively in structures where low weight and high stiffness are required. Their use in the industrial application has been increasing since their first use in the military sector in the 1940s (Mazumdar, 2004).

These materials appear in nearly all areas of the market, from the aerospace and defense business, which is where 52% of the composite industry does most of its work, to automation (33%), wind energy (13%), etc (Sloan, 2016).

A good indicator of its success is the global demand of carbon fiber (CF), which has shown an annual growth rate of approximately 10% per year since 2010, and it is predicted to keep stable over several years (Kühnel and Kraus, 2016).

Structures made of laminates are usually thin-walled and can take complex geometries. Thus, tools are necessary so that designers can model the anisotropic material and complicated geometries to exploit the properties of the material to its full extend. The usual analytical methods are very limited to determine the response of such structures, therefore numerical tools are required.

Numerical methods tendency towards laminated composite structures and MUST

Together with the growth of the use of composite materials in the industrial sector, numerical methods have seen a rapid development over the last decades. Authors like Hughes, Bathe, McNeal, Klinkel, Simo, Rifai and Reddy have contributed significantly to the growth of finite element theory and some stabilization techniques used in this work.

The Department of Materials and Production at Aalborg University has been developing the design tool called MUltidisciplinary Synthesis Tool (MUST), which is finite element based, since 1998. Among a wide spectrum of capabilities, much work has been put into implementing methods for laminate analysis and optimization. This master thesis focuses on the segment related to modeling thin-walled laminated structures through the use of shell finite elements.

MUST is a system developed by Erik Lund and co-workers since 1998. It serves as a tool for design, analysis and optimization of structural, fluid and thermal problems. MUST relies on commercial software for meshing and general preprocessing. The MUST platform contains the implementation of many shell finite elements. The first work on shell elements was performed by Jensen et al. (2001) and further work has been carried out by Stegmann (2005), Hansen and Hvejsel (2007) and Greve et al. (2008).

The shell elements implemented in MUST are generally isoparametric ranging from 4 to 9 nodes. Some of the elements have been enhanced with the Enhanced Assumed Strains (EAS) method. Some others use the Mixed Iterpolation of Tensorial Components stabilization (MITC), a specific case of the Assumed Natural Strains (ANS), which makes them loose their isoparametry. They are implemented with five degrees of freedom (DOF) per node but by means of a penalty stiffness the sixth DOF is achieved.

At this point, being aware of the importance of being capable of correctly modeling laminated composite structures and knowing which softwares are currently available, the motivation of this thesis is found: The implementation of a computationally efficient 4-node shell accurate and robust element to be used for structural laminate optimization.

1.1 Problem statement and solution strategy

The objective of this thesis and the problem to be solved is developing an effective and computationally cheap 4-node shell element that has not yet been implemented in MUST. As explained in the previous lines, MUST platform contains shell elements that are generally isoparametric. Some of them are enhanced in-plane by means of the EAS method and others get a correct out-of-plane bending behavior by stabilizing the element through the MITC methodology.

In this work, a 4-node shell element is coded in Fortran90, and enhanced with both the EAS method and MITC so while being a computationally efficient element, it behaves properly under in-plane bending loading and transverse bending situations. This element must consist of a set of subroutines implemented in the MUST system capable of performing geometrically non-linear structural analyses, linear static analyses and linearized buckling analyses.

The solution of the problem will take offset in a 3^{rd} semester project from the master's degree in Design of Mechanical Systems (DMS) at Aalborg University. In this project the linearized version of the element for plates made of isotropic materials was coded and tested in Matlab, taking offset in the degeneration of a solid 8-node element and improving the isoparametric formulation with the EAS and MITC enhancements to avoid the shear locking phenomenon.

In a nutshell, the objective of this project is implementing in the MUST platform a 4-node shell element for analysis of laminated composite structures capable of solving linear static structural problems, linear buckling and geometrically non-linear problems involving large displacements and small rotations. Such element needs to be enhanced in order to improve the in-plane and transverse shear locking phenomenon. It must be robust, accurate and stable in order to simplify the mesh of thin-walled structures.

1.2 Outline of the project

The thesis report is divided into a number of chapters and sections. It proceeds as follows:

- In chapter 2 an overview on analytical theories for modeling laminated composites is given. Subsequently a short presentation of FE ways to model laminates is done.
- A linear isoparametric quadrilateral shell element suffers from shear locking. This concept is introduced in chapter 3 as well as the MITC method, a solution to avoid the out-of-lane locking.
- In chapter 4 continuum mechanics is derived in curvilinear coordinates. Prior to the derivation, some basic concepts of tensor calculus are introduced so the reader has the appropriate mathematical background to understand the derivation that follows.
- At this point of the thesis, it is where the non-linear enhanced formulation of the element is derived. Chapter 5 formulates the governing equations of the 4-node enhanced shell element by means of calculus of variations.
- Chapter 6 presents the finite element formulation applied in order to solve for the governing equations derived in the previous chapter. The different stiffness matrices are explained and later on their numerical implementation defined.
- In chapter 7 numerical tests are taken in order to check for accuracy and convergence of the linear formulation. In the case of the non-linear element, special benchmarks are studied and compared to ANSYS to assure the performance of the element and the solver are appropriate. The same philosophy is applied to the linearized buckling tests.
- Chapter 8 presents the conclusions of the master thesis.

Brief overview of laminate composite analysis 2

This chapter gives an overview of some general ways to model and analyze laminates made of composite materials. Firstly, an elasticity approach is explored by the means of the shell theories. Later, using the same assumptions but focusing on plates for the sake of simplicity, a review on how these theories can be applied to laminated plates through the so called Single Layer Theories is done. This includes an overview of coupling effects: the behavior of the laminates depending on the layup of the fibres. Finally, a short presentation of the different finite element options to model laminates is done with focus on advantages and disadvantages, and disregarding the element formulation insights.

2.1 Shell theories

In order to obtain a solution to the elastic problem for shells some assumptions need to be taken, similarly to what is done for beams or rods. These assumptions are what differentiate the different shell theories. This section outlines the assumptions, applicability and advantages of the two simplest theories: the Kirchoff-Love and the Mindlin-Reissner assumptions based theories. This short note on these two theories serve as introduction. If the reader is interested in complete derivations and discussions on these two theories the following references might be of interest: Reddy (2009) gives a complete picture of both geometry and linear elasticity of shells while Timoshenko and Woinowsky-Krieger (1987) shows different examples of plates and shells under different loading conditions.

2.1.1 Classical Shell Theory

The classical shell theory is based on the Kirchoff-Love hypothesis for plates and shells, that are an extension of the ones made in the Euler-Bernoulli beam theory:

1. Transverse normals are inextensible $\Rightarrow \varepsilon_{tt} = 0$.
2. Transverse normals remain straight after deformation.
3. Transverse normals remain normal to the middle surface of the shell after deformation $\Rightarrow \gamma_{rt}, \gamma_{st} = 0$.

Thus, using the Cauchy strain tensor the different assumptions yield the following results:

$$\varepsilon_{rr} = \frac{\partial w}{\partial t} = 0 \Rightarrow w(r,s,t) = w_0(r,s) \quad \text{Because of assumption 1} \tag{2.1}$$

$$\gamma_{rt} = \frac{\partial u}{\partial t} + \frac{\partial w}{\partial r} = 0 \Rightarrow u(r,s,t) = u_0(r,s) - t\frac{\partial w_0(r,s)}{\partial r} \quad \text{Because of assumption 3} \tag{2.2}$$

$$\gamma_{st} = \frac{\partial v}{\partial t} + \frac{\partial w}{\partial s} = 0 \Rightarrow v(r,s,t) = v_0(r,s) - t\frac{\partial w_0(r,s)}{\partial s} \quad \text{Because of assumption 3} \tag{2.3}$$

where all the displacements are local displacements in the position (r, s, t) and $(u_0.v_0, w_0)$ is the displacement of the middle surface or reference surface. The displacements are represented in figure 2.1 where the assumptions done can be observed.

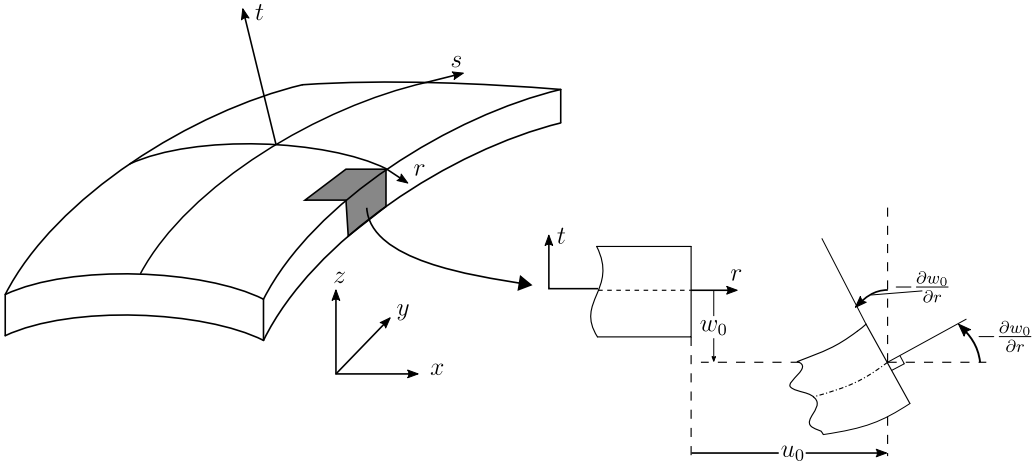


Figure 2.1. Kirchoff-Love assumptions represented in a deformed shell

From this displacement field, the strains and therefore the stresses can be calculated as it is seen in section 2.2.

The basic limitations of the classical shell theory are similar to the ones in the Euler-Bernoulli beam theory and therefore, the results are only accurate for low curvature and low thickness to in-plane dimension ratio. Regarding laminated shells, the classical shell theory yields unsatisfactory results if the in-plane stiffness is significantly higher than the transverse stiffness because relatively high transverse shear strains can occur that are not accounted for in the Kirchoff-Love assumptions.

One of the main advantages of the classical shell theory is its simplicity and that only depends on three variables. However, the price to pay for the simplicity is having to calculate the derivatives wrt. r, t of the middle surface displacement w_0. This is a drawback if an implementation in numerical methods such as the finite element method is done, because second order interpolation functions need to be implemented in the thickness direction to ensure differentiability making the model less efficient.

Another problem, consequence of not taking into account the transverse strains, is that the stresses responsible of the delamination failure mode are not taking into account. As stated in Jones (1998), such stresses are the highest in the edges while in the middle of shells and plates this effects are small or non-existent. In order to account for them, equilibrium equations can be set up that will, through an integration process, give the transverse shear stresses and if needed the transverse normal stress.

As a consequence of this disadvantages, the finite element formulations used to model laminated shells usually use the Reissner-Mindlin shell theory.

2.1.2 Rissner-Mindlin shell theory

The formulation of the Reissner Mindlin theory follows the same procedure as the classical shell theory but with a change on the assumptions. In this case the assumptions are adapted from

the Timoshenko beam theory:

1. Transverse normals are inextensible $\Rightarrow \varepsilon_{tt} = 0$.
2. Transverse normals remain straight after deformation.
3. Transverse normals are allowed to rotate with angles $\phi_r(r,s) = \frac{\partial u}{\partial t}$, $\phi_s(r,s) = \frac{\partial v}{\partial t} \Rightarrow \gamma_{rt}, \gamma_{st} \neq 0$.

Following the same procedure as for the classical shell theory, the displacement field can be expressed with the help of the assumptions as:

$$\varepsilon_{rr} = \frac{\partial w}{\partial t} = 0 \Rightarrow w(r,s,t) = w_0(r,s) \quad \text{Because of assumption 1} \tag{2.4}$$

$$\gamma_{rt} = \frac{\partial u}{\partial t} + \frac{\partial w}{\partial r} \neq 0 \Rightarrow u(r,s,t) = u_0(r,s) + t\phi_r(r,s) \quad \text{Because of assumption 3} \tag{2.5}$$

$$\gamma_{st} = \frac{\partial v}{\partial t} + \frac{\partial w}{\partial s} \neq 0 \Rightarrow v(r,s,t) = v_0(r,s) + t\phi_s(r,s) \quad \text{Because of assumption 3} \tag{2.6}$$

Again, from this displacement field, the strains can be calculated. The quantities defined in the above equations can be visualized in figure 2.2.

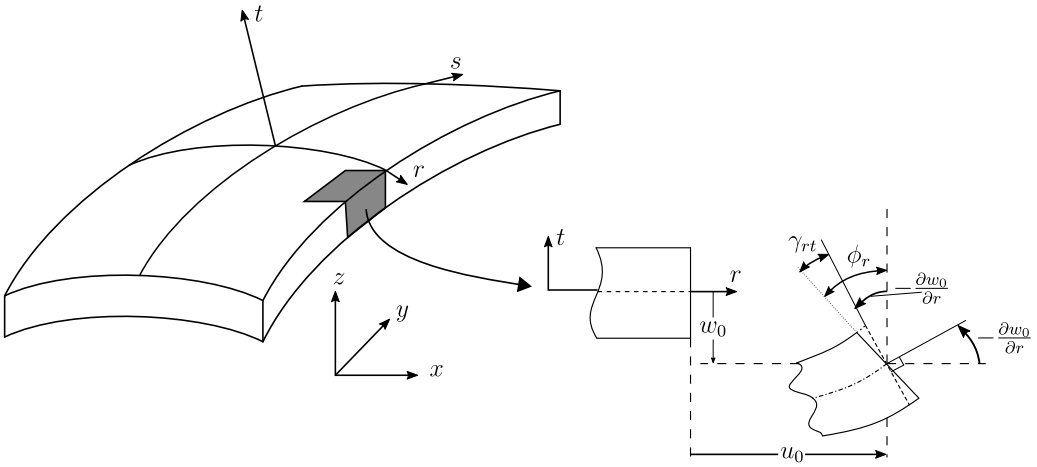


Figure 2.2. Kirchoff-Love assumptions represented in a deformed shell

Rotations ϕ_r and ϕ_s do not represent rotations around r and s axes that follow the right hand rule. To solve this problem a redefinition of them is done that will be used for the rotational degrees of freedom of the element:

$$\beta_r = -\phi_s, \qquad \beta_s = \phi_r \tag{2.7}$$

Unlike the classical shell theory, the Mindlin-Reissner theory provides a good result for thin and moderately thick shells. The numerical codes that are formulated with this theory are simpler to implement because lower order derivatives are needed. The price to pay for this simplicity is, however, the addition of two rotational degrees of freedom α and β. In addition, transverse shear strains and stresses are available for their use in predicting through-the-thickness failures such as delamination. The transverse normal stress is not available because of the normal non-extension assumption.

The transverse shear stresses are piecewise continuous and, therefore, do not satisfy equilibrium on the lamina interface. The most important problem however, comes from the fact that

transverse shear stresses are not parabolic. This produces an inaccurate shear strain energy and therefore, an inaccurate result in a finite element model. This is corrected using a shear correction factor, that multiplies C_{55} and C_{66} in the constitutive matrix $[C]$, which does not change the shape of the stress distribution but allows the proper calculation of the shear strain energy value as if the distribution was parabolic. The shear correction factor is problem dependent and actually very difficult to calculate in the case of composite material laminates. For isotropic materials the factor is more or less constant and it is usually set to $=\frac{5}{6}$. Because of the complexity of the problem, the choosing of the shear correction factor is set to $0,8$ in this work. However, if the reader is interested in some examples of how the shear factor is calculated can be found in Timoshenko (1922) where the shear correction factor is proved to be dependent on the Poisson's ratio of the material. In Cowper (1966) the topic is treated more deeply and generally for the case of the Timoshenko beam theory. Examples for different laminate types plates and shells can be found in Vlachoutsis (1992).

2.2 Equivalent single layer theories

The solution of the elastic problem of laminated shells using the theories outlined in section 2.1 result in the equivalent single layer theories for laminated shells and plates. Because of the geometry of plates is easier than the one for shells, in this section, the theories are outlined for plates, although the final results are similar. The formulation of such theories can be found in Jones (1998) for plates or Reddy (2009) for general double curved shells. In the following lines only an outline of the derivation, the results and its implications are presented.

The basic idea behind the derivation of equivalent single layer theories is to split the strains in a component depending on the middle surface strains and the middle surface curvatures terms. Depending on which assumptions are used in order to define the curvature associated terms, the two main equivalent single layer theories are obtained. If the classical shell theory is followed, the curvature terms are the real curvature of the shell and the classical laminated plate theory is obtained. If the Mindlin-Reissner shell theory is used, the curvature terms are related to the derivatives of the angles α and β and the First Shear Deformation theory for plates (FSDT) is obtained.

Then, the force, moments and shear (in the case of the FSDT) resultants are expressed in terms of the integrals of the stress filed. As the stress field is a function depending on the strains and the mechanical properties of each layer k, the integrals can be calculated as weighted summations. The final result is expressed in the form of an A-B-D matrix in the case of the Classical Laminate Theory, as seen in (2.8).

$$\left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \\ \hdashline M_x \\ M_y \\ M_{xy} \end{array}\right\} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ \hdashline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{array}\right] \left\{\begin{array}{c} \varepsilon^{\circ}_{xx} \\ \varepsilon^{\circ}_{yy} \\ \gamma^{\circ}_{xy} \\ \hdashline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array}\right\} \tag{2.8}$$

where each coefficient is obtained by

$$A_{ij} = \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k - z_{k-1}\right), \quad i,j = 1,2,6 \quad \text{(extension stiffness)}$$

$$B_{ij} = \frac{1}{2}\sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^2 - z_{k-1}^2\right), \quad i,j = 1,2,6 \quad \text{(bending-extension coupling stiffness)}$$

$$D_{ij} = \frac{1}{3}\sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^3 - z_{k-1}^3\right), \quad i,j = 1,2,6 \quad \text{(bending stiffness)}$$

here, N is the number of layers, and z is the coordinate of the lamina as explained in Jones (1998, pp. 197) and $[\bar{Q}]$ is defined as:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix}_k = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{21} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{31} & \overline{Q}_{32} & \overline{Q}_{66} \end{bmatrix}_k \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \tag{2.9}$$

which is the constitutive relation for a lamina expressed in the laminate coordinate system.

For the FSDT, the $[\bar{Q}]$ matrix is expanded to be able to express the transverse shear strains and the A-B-D matrix is completed with an $[\bar{A}]$ term that expresses the transverse shear terms. The following explanation about the coupling effects is not affected by which theory is used and, therefore, the results for the FSDT are not displayed for the sake of simplicity. Such results can be found in Reddy (2009, sec. 3.4.3).

Looking at the A-B-D matrix it can be seen that for a general laminate there are several coupling effects:

- Shear-extension coupling: Terms A_{16} and A_{26}.
- Bending-extension coupling: $[B]$ matrix.
- Bend-twist coupling: Terms D_{16} and D_{26}.

The coupling behavior can be of extreme importance for the manufacturing process and for the properties of the part being designed. Therefore, it is important to test if the finite element formulated in this work displays such phenomena appropietely. To ensure that, in chapter 7 this phenomena is tested. An extense explanation on the coupling phenomenon with several examples can be found in Jones (1998).

It is also worth to mention that multi-layer approaches can also be followed as explained in Reddy (2009). In order not to extend this introductory section these theories are left out.

2.3 Layered finite elements

When using finite elements to model laminated shell structures mainly two groups of elements exist: solid shell continuum elements, that use a 3D formulation, or degenerated shell elements that use a 2D double curved formulation. Another subgroup in the degenerated shell elements would be the basic plate elements that can be used as facets to model surfaces or to directly model plates without curvature. Another division between single layer equivalent and multiple layer elements can be done. However, as the objective of this work is to formulate an element that

can be used in a general optimization procedures for structures such as wind turbine blades, the multiple layer elements are not explored as an option because of their complexity and computational resources requirements.

The plate and shell elements were formulated in the 1960's and have greatly evolved since then. In the single layer shell elements an important enhancement to the 2D double curved formulation was achieved in Dvorkin and Bathe (1984) and Dvorkin and Bathe (1985) when the Mixed interpolation Tensorial Components (MITC) were introduced. As a result, the shear locking phenomena was reduced and a more reliable but still cheap 4-node shell element was formulated.

In the late 1990's the enhancing techniques evolved to a point in which 3D continuum mechanics based elements could be adapted for the modeling of the shell. This meant solving the problem of shear locking in the transverse and in-plane directions as well as the volumetric locking for higher order elements. A decisive contribution was done in Simo and Rifai (1990) regarding the incompatible modes and Enhanced Assumed Strain method (EAS) used in this work. The EAS method was applied in Klinkel and Wagner (1997) for the 4-node shell element with good results and later on, combined with the MITC approach, in Klinkel et al. (1999) in a solid shell 3D element. Nowadays, new formulations based on improving some of the still bad behaviours of both shell and solid shell elements like the MITC4+ element in Ko et al. (2016), have been developed by doing slight changes in the existing formulations.

When choosing between the wide variety of shell and solid shell elements a compromise between the computational performance, the analysis requirements and the structure to model needs to be done. Then for example, solid shell elements offer good approximations for the through-the-thickness stresses but are generally more computationally expensive. Two-dimendional based shell elements are slightly computationally cheaper and can model surfaces with simpler meshes but the through the thickness stresses are less accurate. Generally, degenerated shell elements are preferred for large structures or rough analysis to later on, use more expensive elements to evaluate critical areas where, for example, delamination could occur.

Shear Locking and its solutions 3

Shear locking is a problem that arises in shell elements due to the discretisation of the domain done when formulating elements. In this section it is assumed that the reader has a basic knowledge of the finite element method. In the case of a short review of some basic isoparametric formulation, appendix A has a short summary of it. In this section the locking effect in general is described with emphasis on the shear locking. Lastly, solutions to the shear locking are proposed focusing on the ones used in this work: Mixed Interpolation of Tensorial Components (MITC) and Enhanced Assumed Strains (EAS).

3.1 Locking phenomena

Locking phenomena arises in finite elements when the interpolation chosen to express the displacement field of an element is unable to model the real deformation of the structure, making the model become over-stiff. As a consequence, the analysis results can be inaccurate or even completely incorrect. There are mainly three locking phenomena: shear locking, membrane locking and volumetric locking.

3.1.1 Volumetric locking

The volumetric locking is presented first because it is not exactly caused by the displacement interpolation. Volumetric locking arises because of the material having a Poison ratio close to $0,5$. This effect can be understood through the expression of the Lamé parameters in the generalized Hooke law for isotropic materials:

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + +2\mu\varepsilon_{ij} \tag{3.1}$$

being the Lamé constants defined as $\lambda = \frac{E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$.

If the Poison ratio is close to 0.5, λ tends to infinity, the stresses become big and therefore less strain is required to reach the stationary point in (A.2). Another way to understand it, is by means of the bulk modulus which expresses the compressibility of a material and also tends to infinity making the material incompressible. If the elements are incompressible no deformation can happen and therefore the mesh will lock.

Volumetric locking can occur in orthotropic materials if the poison ratio tends to 1. However, as composite materials are mostly used in laminates, which combine their mechanical properties, it is extremely difficult, if not impossible, that a combination of common linear elastic orthotropic materials can lead to a combination of mechanical properties producing volumetric locking. As the element formulated in this work is intended for analysis of "common" laminated structures, the volumetric locking problem is not addressed.

3.1.2 Membrane locking

Membrane locking is a locking phenomena that appears in curved shell and beam elements. Membrane locking occurs when these structures are not capable to properly reproduce an inextensional bending situation. Such situation is produced in shells and slender beams because the bending stiffness is much lower than the membrane stiffness. As demonstrated in Cook et al. (2001, pp. 567) with a curved beam element, the problem only appears for small radius of curvature. The locking is produced because artificial membrane strains appear that, due to the high membrane stiffness, produce high membrane stresses. The only way that the structure has to achieve a stationary point for the potential energy is reduce the strains, making the structure to lock.

As this problem only appears for curved elements, that are of higher order than the 4-node shell element, this locking phenomenon is not a problem for our formulation. However, the EAS or MITC method would in some cases successfully solve this problem.

3.1.3 Shear locking

Shear locking occurs when the interpolation chosen for the element is not able to represent pure bending situations. As with the membrane locking, this creates spurious strains that spend energy that should be spent in the bending deformation. As this is the locking problem that needs to be solved in the case of the 4-node shell element of this work, it is explained in more detail in the following lines.

Mathematically, this parasitic shear strain appears when computing the strains from the displacement interpolations. Considering the in-plane case of the 4-node example element formulated in appendix A, the displacement field is ruled by the 2D bilinear shape functions introduced in section 6.1, equation (6.4). Thus, the in-plane displacements u and v are expressed as:

$$u = \sum_{I=1}^{4} N_I^{2D} u_I \quad \text{and} \quad v = \sum_{I=1}^{4} N_I^{2D} v_I \tag{3.2}$$

With these interpolations, the element will behave as in figure 3.1 (b) and the kinematics of the element in pure bending are:

$$u = \sum_{I=1}^{4} N_I^{2D} u_I = rs\bar{u} \quad ; \quad v = \sum_{I=1}^{4} N_I^{2D} v_I = 0 \qquad \text{being} \qquad r = \frac{x}{a} \, , \, s = \frac{y}{b} \tag{3.3}$$

This deformation state yields the followings strains:

$$\varepsilon_x = \frac{\partial u}{\partial x} = s\frac{\bar{u}}{a} \, , \, \varepsilon_y = \frac{\partial v}{\partial y} = 0 \, , \, \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = r\frac{\bar{u}}{b} \tag{3.4}$$

It is noticed that there is no Poisson effect. Therefore, the normal strains in the y direction are zero. Besides, parasitic shear strains γ_{xy} are inherited due to a zero vertical displacement. See Cook et al. (2001, pp. 98-100) and Lund and Lindgaard (2016, lec. 2) for deeper explanation and derivation.

The transverse shear locking, for the case of the 4-node shell element formulated in this work, is a similar case in which the element locks due to the presence of spurious transverse shear strains γ_{xz} and γ_{yz}.

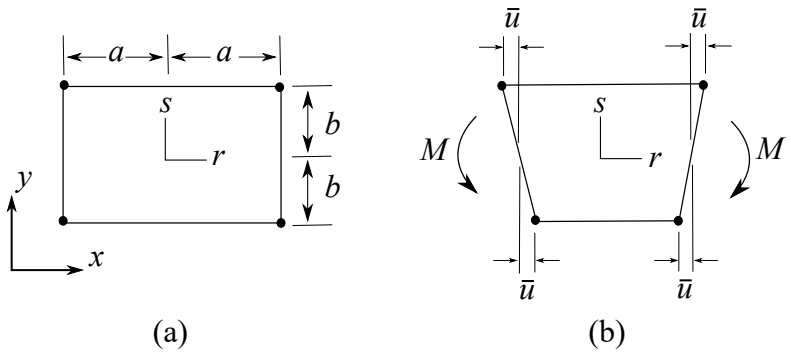


Figure 3.1. (a) Top view of the rectangular 4-node shell element. (b) 4-node shell element deformed by pure in-plane bending moment.

The solution to solve the shear locking is to use tricks that eliminate the spurious shear strains. In order to do so, different strategies are possible. It is known that for a pure bending case shear strains should be 0, evaluating the strains in places where the strains are 0 and introduce them in the strain displacement matrix is a way to fix the shear locking problem. From this approach two main methods can be found. The first one is to use selective integration points in which the strains are known to be zero or low. From another point of view, by producing a numerical error on the calculation it corrects the error produced by the shear strains. This has also the advantage of using less integration points and, therefore reducing the computational cost of the element. However, the hour-glass problem is introduced that can produce extremely inaccurate results. In order for the elements with selective or reduced integration to work, it is necessary a stabilization process that avoids the hour-glass behavior.

Another way to solve it is to introduce internal degrees of freedom that produce displacement fields that can represent pure bending. This would be like increasing the order of the element but at a much cheaper cost. These methods are called incompatible modes methods because the displacement field is incompatible and elements overlap. In order for these formulations to work, some extra steps need to be done to ensure that the element can pass the patch test. A subgroup of this methods are the so called Enhanced Assumed Strains (EAS) that do the same but with the strain field instead of doing it directly to the displacement field.

The Enhanced Assumed Strain (EAS) is better understood through the formulation of the element itself. However its basic idea is to introduce an incompatible strain field to alleviate the locking effect. The formulation is carefully done in order to past the patch test. The last group of methods to solve the shear locking are the so called B-bar methods that modify the interpolations in the strain displacement $[B]$ matrix. The MITC, that is enclosed in the so called Assumed Natural Strains group of methods, is explained in the following section because it is the approach chosen for the 4-node shell element transverse shear strains enhancement.

3.2 Mixed Interpolation of Tensorial Components

The MITC formulation proposed in Dvorkin and Bathe (1984) and Dvorkin and Bathe (1985) is widely used in different element formulations and it is the one proposed for the 4-node shell element. In this formulation, the transverse shear strains are re-interpolated from four chosen tying points as shown in figure 3.2.

The reason for choosing these points is the same as explained in the last section: they give

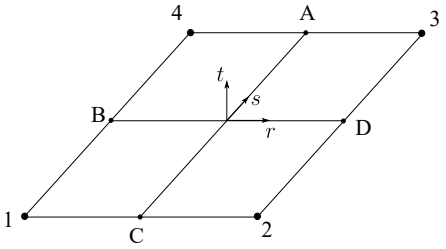


Figure 3.2. Representation of a 4-node shell element with its tying points.

good results in case of a pure bending situation. In figure 3.3 it can be seen that in the centre of the element profile the shear strain when there is a pure bending situation is zero. This means that any point in the lines AC or BD is suitable to be a good tying point. In order to obtain a better and simpler interpolation between the pair of tying points, these are placed in the edge of the element.

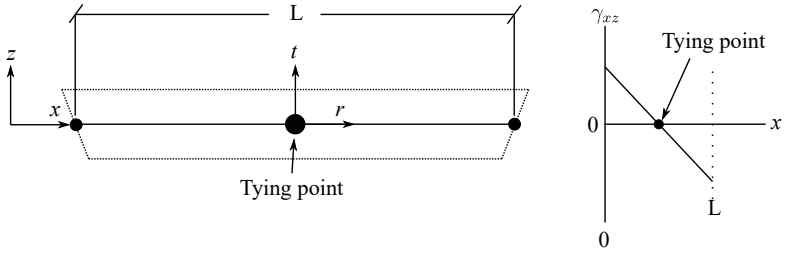


Figure 3.3. Transverse shear strain calculation for a single element in a pure bending situation.

Then, for the shell elements, the transverse shear strains are assumed to be constant along the thickness of the element following the assumptions of the Reissner-Mindlin shell theory. Using the interpolations shown in figure 3.4:

$$N_A^{rt} = \frac{1}{2}(1+s), \qquad N_C^{rt} = \frac{1}{2}(1-s) \tag{3.5}$$

$$N_B^{st} = \frac{1}{2}(1-r), \qquad N_D^{st} = \frac{1}{2}(1+r) \tag{3.6}$$

the strains can be expressed as:

$$\begin{aligned} \tilde{\gamma}_{st} &= \frac{1}{2}(1+r)\tilde{\gamma}_{st}^D + \frac{1}{2}(1-r)\tilde{\gamma}_{st}^B \\ \tilde{\gamma}_{rt} &= \frac{1}{2}(1+s)\tilde{\gamma}_{rt}^A + \frac{1}{2}(1-s)\tilde{\gamma}_{rt}^C \end{aligned} \tag{3.7}$$

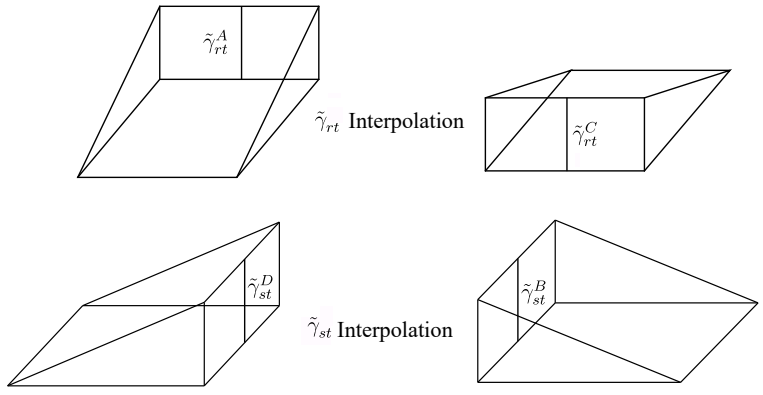


Figure 3.4. Shape functions used to interpolate the transverse shear strains.

The numerical implementation of the MITC can be seen in chapter 6. The main problem of the EAS and the MITC are that the interpolations are thought for rectangular shaped elements. When distortion is applied to the elements the results become inaccurate as it is seen in chapter 7. Therefore it is important to make sure that the mesh quality is sufficiently high.

Continuum mechanics in curvilinear coordinates 4

Laminated shell elements are often used to analyse double curved shells made of orthotropic materials. Therefore, it is important to introduce some concepts related to the formulation of the elasticity equations in curvilinear coordinate systems. In order to do so, some mathematical concepts are introduced and references given to more extensive derivations, as the topic is too extense to treat in depth in this work.

4.1 Basic concepts of tensor calculus

Tensor calculus is a broad field of mathematics that is impossible to convey in this work and consequently, this section is not intended to develop tensor calculus as a whole but provide a set of useful tools for continuum mechanics in curvilinear coordinate systems. Most of the tensor calculus concepts needed in continuum mechanics are well defined in books such as Dym and Shames (2013), Bathe (1996) or Rudnicki (2014). However if the reader wants more information about the matter in a more mathematical wise structured way the authors recommend Fleisch (2012) for a general introduction and Synge and Schild (1978) or Heinbockel (1996) for more extensive derivations and concepts.

4.1.1 Preliminary definitions

In mathematics, tensor is a broad term that includes quantities that do not vary, independently of the base chosen to represent them. In Fleisch (2012) there is a definition of tensor that conveys this idea of invariability:

"A tensor of rank n is an array of 3^n values (in 3-D space) called "tensor components" that combine with multiple directional indicators (basis vectors) to form a quantity that does not vary as the coordinate system is changed."

This can be clearly seen in the case of tensors of rank one and zero, scalars and vectors respectively, with a physical quantity such as velocity, that does not change if the base is changed. In other words, the representation of this velocity will change but not the velocity itself. Index notation is often used in this work in order to express tonsorial equations in a shorter way. For a first rank tensor the notations is as follows:

$$\{v\} = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} = v_i = v_x\{e_1\} + v_y\{e_2\} + v_z\{e_3\} \tag{4.1}$$

Here $\{e_1\}, \{e_2\}, \{e_3\}$ are the standard base vectors and v_x, v_y, v_z are the vector components.

In this work, tensors of rank two are of the most importance because they are used to represent strain and stress. Second order tensors can be expressed both as 3x3 matrices or by making use of indicial notation:

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = A_{ij} \tag{4.2}$$

In order to express the differentiation of both quantities comma notation is used in this work:

$$v_{i,j} = \{v\}_{,j} = \frac{\partial\{v\}}{\partial x_j} = \frac{\partial v_i}{\partial x_j} \qquad A_{ij,k} = [A]_{,k} = \frac{\partial[A]}{\partial x_k} = \frac{\partial A_{ij}}{\partial x_k} \tag{4.3}$$

Another aspect of the index notation worth reminding is the difference between repeated and dummy index, as it is widely used in some sections of this work. In index notation the so called Einstein summation convention is used, implying that a repeated index implies a summation, making expressions much more compact.

$$\sum_{i=1}^{n} a_{ij} b_i = a_{1j} b_1 + a_{2j} b_2 + ... + a_{nj} b_n = a_{ij} b_i \tag{4.4}$$

In the expression above, the repeated indices are also called dummy indices while the indices that are free to take any allowed value (j in this case) are called free indices.

4.1.2 Tensor transformation, covariance and contravariance

It has been mentioned that what makes the tensors so useful in mathematics and physics is the ability to transform from one base to another without changing the magnitude they represent. The transformation itself does not have a lot of secrets: the tensor itself does not change but its components do. However, the answer to what happens when the reference system is subjected to a transformation and the relation between how tensors and bases transforms yield some interesting and useful results.

It is known that a tensor of rank one can be transformed by the use of a transformation matrix $[T]$ and it is a requirement that this transformation is reversible by the use of $[T]^{-1}$. The expression used is then $V' = [T]V$. It is also important to note that this transformation can also be used for non-orthonormal bases without changing the transformation rule.

If a rotation is performed to a 2-D vector the following rule applies:

$$\begin{pmatrix} V'_x \\ V'_y \end{pmatrix} = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \end{pmatrix} \tag{4.5}$$

However, if what needs to be rotated are the base vectors the following rule applies:

$$\begin{pmatrix} e'_{ix} \\ e'_{iy} \end{pmatrix} = \begin{pmatrix} cos(\alpha) & sin(\alpha) \\ -sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} e_{ix} \\ e_{iy} \end{pmatrix} \tag{4.6}$$

It can be seen that the transformation matrix for the vector in equation (4.5) is the inverse of the axis transformation in equation (4.6). For this reason, the components of the vector are called

contravariant components. The question that arises after that is whether the vector also has another set of components called covariant in which the same transformation is used for both vectors. In order to answer this question and also to generalize some concepts, it is useful to move to non orthonormal basis.

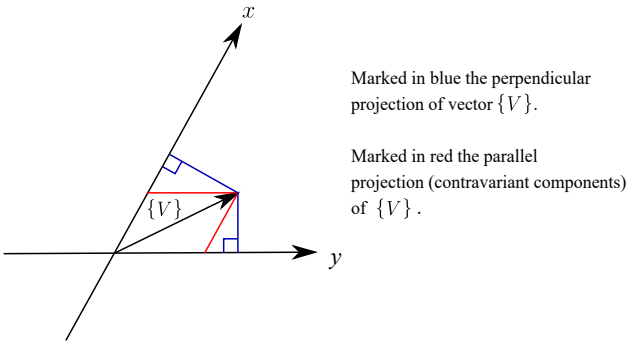


Figure 4.1. Two possible projections of a vector $\{V\}$ in a non orthonormal coordinate system

As can be seen in figure 4.1, there are two possible projections of the vector $\vec{V}$ to the axis x, y depending on whether the projection is done perpendicular to the axis or parallel to the other axis. This determines the components of the vector and will defer depending on what approach is taken. It is proved in Fleisch (2012) that only the parallel approach (contravariant components) works when vector addition is introduced.

To be able to work with the covariant components, another base called reciprocal base or contravariant base is needed, as can be seen in figure 4.2.

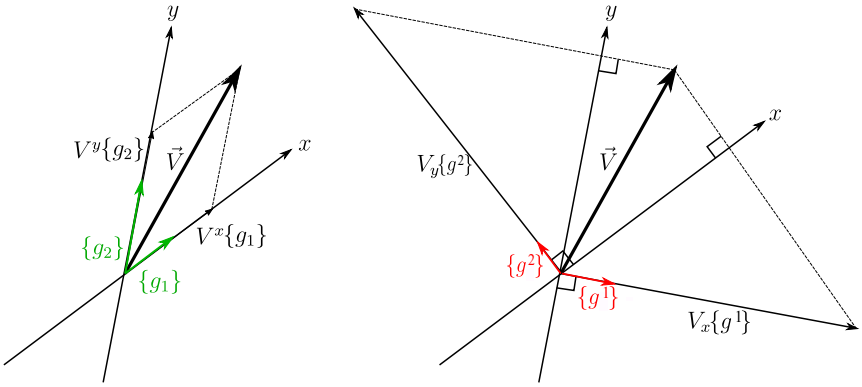


Figure 4.2. Left: Expression of $\{V\}$ using the covariant base and the contravariant components of the vector. Right: Expression of $\{V\}$ using the contravariant base and the covariant components of the vector.

The covariant base is always tangent to the non orthonormal axis while, as seen in the figure 4.2, the contravariant base then has to fulfil two requirements:

- Each contravariant base vector has to be perpendicular to all original basis vectors with different indices
- The dot product between each contravariant basis vector and the original basis vector, that has the same index, has to be one.

So far a description of the covariance and contravariance has been given. However, the advantages of using such a formulation have not been explained. The main advantage of using covariant and contravariant components is that when combined adequately they are not affected

by transformations of the basis vectors. As an example, the vector $\{V\}$ in figure 4.2 can be expressed in an orthonormal base as:

$$\{V\} = V^x\{g_1\} + V^y\{g_2\} = V_x\{g^1\} + V_y\{g^2\} \tag{4.7}$$

It is known that contravariant quantities use an inverse transformation matrix when transformed, while covariant quantities use a direct transformation rule. This means that when the two kinds of components are combined, the quantity being described does not vary when a transformation is applied.

4.1.3 Advanced tensor operations

Tensors are tools that help characterize physical or mathematical magnitudes. Therefore, it is necessary to define some operations that allow the use of these tools to solve problems. Although most of these operations make more sense when they are used, they are presented here to have a quick reference for the theoretical derivations presented in this work. The term "advanced" in the title of this section does not refer to the difficulty of the operations presented here but to the fact that they are not commonly used and they are worth a reminder.

Outer product

The outer product of two vectors is obtained when multiplying a column vector by a row vector in the following manner:

$$[C] = \{A\} \otimes \{B\} = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} \begin{pmatrix} B_1 & B_2 & B_3 \end{pmatrix} = \begin{pmatrix} A_1B_1 & A_1B_2 & A_1B_3 \\ A_2B_1 & A_2B_2 & A_2B_3 \\ A_3B_1 & A_3B_2 & A_3B_3 \end{pmatrix} \tag{4.8}$$

In index notation it is expressed as:

$$c_{ij} = a_i b_j \tag{4.9}$$

Although this result may seem meaningless it will prove to be useful in section 4.2.1. Some important properties about this product are:

- Not commutative $\{a\} \otimes \{b\} \neq \{b\} \otimes \{a\}$
- If $\{a\} \otimes \{b\} = [A]$ being $[A]$ a matrix, the following proves: $[A]\{b\} = \{a\}\|\{b\}\|^2$

Colon product

Colon product can be seen as an extension of the scalar product of two vectors for general tensors. It is defined as follows:

$$[a] : [b] = a_{ij}b_{ij} = a_{11}b_{11} + a_{12}b_{12}... \qquad [c] : [d] = c_{ij}d_{ijk} = \begin{pmatrix} c_{11}d_{111} + c_{12}d_{121}... \\ c_{11}d_{112} + c_{12}d_{122}... \\ \vdots \\ c_{11}d_{11k} + c_{12}d_{12k}... \end{pmatrix} \tag{4.10}$$

In expression (4.10) the expression on the left corresponds to the multiplication of equal rank tensors while, the expression on the right corresponds to the multiplication of different rank

tensors, that will give a tensor of rank $rank(a) - rank(b)$. One has to be careful when using the colon product as two definitions exists. The one presented in (4.10) is known as Reddy definition. Holzapfel definition holds as $[a] : [b] = a_{ij}b_{ji}$. When the two tensors that are being multiplied are symmetric (as happens in continuum mechanics for the stress and strain tensors) there is no difference in the final result.

Metric tensor

Although the metric tensor is not an operation itself it is briefly introduced here to have all the tools that are used in section 4.2 together. The metric tensor defines critic or fundamental quantities of the coordinate system chosen and it is defined as:

$$g_{ij} = \{g_i\} \cdot \{g_j\} = g_i^x g_j^x + g_i^y g_j^y + g_i^z g_j^z \tag{4.11}$$

in which the component g_{ij} of the tensor is the scalar product of the two base vectors defining the coordinate system.

Some properties of the metric tensor are important and widely used in this work. The first one can be summarized in the following expression:

$$ds^2 = g_{ij}dx^i dx^j \tag{4.12}$$

where ds is the change in length of a vector $d\{r\} = \{e_i\}dx^i$ describing the infinitesimal distance between two points. The expression (4.12) allows to calculate a change in distance, when a change in the coordinate system occurs, and proves the utility of the covariant and contravariant components and bases, as well as the idea of the metric tensor of representing the critical metrics of the base chosen.

Another interesting property of the metric tensor is the so called index raising or lowering. As has already been seen, a sub-index refers to a covariant component or base while a super-index refers to a contravariant one. The metric tensor can be used to lower or rise the components of tensors:

$$g_{ij}A^j = A_i \qquad g^{ij}A_i = A^i \tag{4.13}$$

where g_{ij} is the inverse of the metric tensor.

4.2 Continuum mechanics in curvilinear coordinates

4.2.1 Strain tensor formulation in curvilinear coordinates

All the concepts mentioned in the last section are used here to define the continuum mechanics concepts needed for the formulation of the shell element. The 4-node shell element formulated in this work must be able to work for general double curved shells. This implies that the continuum mechanics concepts needed have to be adapted to curved geometries.

The position vector $\{X\}$ expresses the position of a point in the global Cartesian coordinate system in the original configuration while vector $\{x\}$ expresses the same in the current configuration. Thus, the covariant base $\{G_i\}$ can be computed as the tangent to the convective system shown in figure 4.3:

$$\{G_i\} = \frac{\partial\{X\}}{\partial x_i} \qquad \{g_i\} = \frac{\partial\{x\}}{\partial x_i} \qquad x_i = r, s, t \tag{4.14}$$

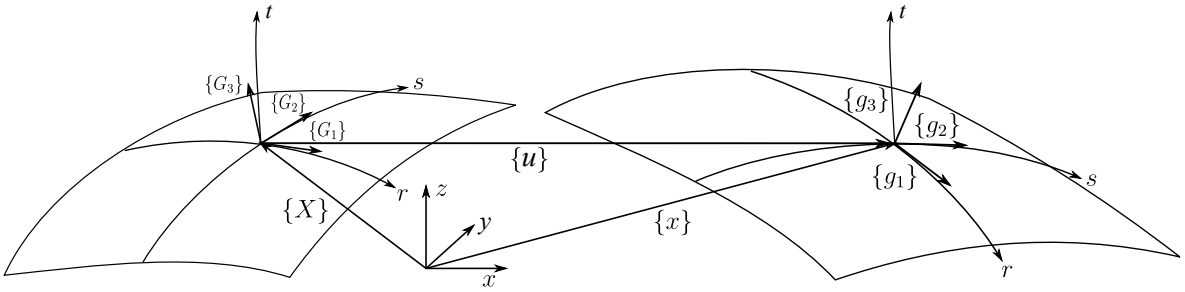


Figure 4.3. Double curved shell with the convective, curved r, s, t coordinate system and the covariant base associated $\{g_1\}, \{g_2\}, \{g_3\}$. Left: Original configuration. Right: Current configuration

The contravariant base vectors $\{G^i\}$ can be calculated by the use of the metric tensor following (4.13) or by simply inverting the derivative of (4.14).

$$\{g^i\} = g^{ij}\{g_j\} \qquad \{g^i\} = \frac{\partial x_i}{\partial \{X\}} \qquad x_i = r, s, t \tag{4.15}$$

Looking at (4.14) and (4.15) it can be noted that the jacobian of the transformation from the global Cartesian space is made by the use of the covariants:

$$[J] = \begin{pmatrix} g_1^x & g_2^x & g_3^x \\ g_1^y & g_2^y & g_3^y \\ g_1^z & g_2^z & g_3^z \end{pmatrix} = \begin{pmatrix} \frac{\partial X}{\partial r} & \frac{\partial Y}{\partial r} & \frac{\partial Z}{\partial r} \\ \frac{\partial X}{\partial s} & \frac{\partial Y}{\partial s} & \frac{\partial Z}{\partial s} \\ \frac{\partial X}{\partial t} & \frac{\partial Y}{\partial t} & \frac{\partial Z}{\partial t} \end{pmatrix} \tag{4.16}$$

In the same way, the inverse of the jacobian contains the contravariants:

$$[J]^{-1} = \begin{pmatrix} g_x^1 & g_y^1 & g_z^1 \\ g_x^2 & g_y^2 & g_z^2 \\ g_x^3 & g_y^3 & g_z^3 \end{pmatrix} = \begin{pmatrix} \frac{\partial r}{\partial X} & \frac{\partial s}{\partial X} & \frac{\partial t}{\partial X} \\ \frac{\partial r}{\partial Y} & \frac{\partial s}{\partial Y} & \frac{\partial t}{\partial Y} \\ \frac{\partial r}{\partial Z} & \frac{\partial s}{\partial Z} & \frac{\partial t}{\partial Z} \end{pmatrix} \tag{4.17}$$

To obtain a formulation that can display large displacements and small rotations, the Green-Lagrange strain formulation will be used. Such formulation compensate the virtual strains produced by rigid body motions that Cauchy linear strain shows. This can be seen if a rotation is applied to a 2-D body.

$$\begin{aligned} x &= Xcos(\phi) - Ysin(\phi) & u &= X(cos(\phi) - 1) - Ysin(\phi) \\ y &= Xsin(\phi) + Ycos(\phi) & v &= Xsin(\phi) + Y(cos(\phi) - 1) \end{aligned} \tag{4.18}$$

Thus, the engineering strain is:

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial X} = cos(\phi) - 1 \\ \varepsilon_{yy} &= \frac{\partial v}{\partial Y} = cos(\phi) - 1 \\ \varepsilon_{xy} &= \frac{1}{2}\left(\frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X}\right) = -sin(\phi) + sin(\phi) = 0 \end{aligned} \tag{4.19}$$

This yields a fictitious strain on x and y directions as illustrated in figure 4.4.

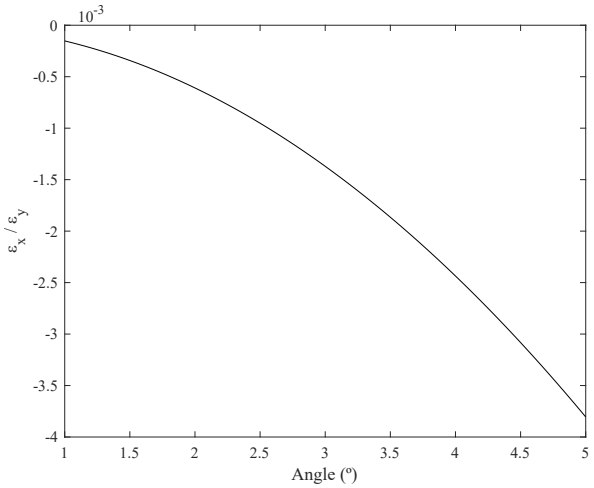


Figure 4.4. Fictitious strains when rigid body rotation is applied.

This error in the strain calculation is remarkable even for small angles. Just to have a reference, some steels start yielding at a value around $0,002$ ε that is achieved in this case for an angle of $3,5°$ (without taking into account any failure criteria) as seen in figure 4.4.

The solution to take out the contribution of the rotations of the strain tensor is the Green-Lagrange strain which adds non-linear terms that cancel out the effects of the rotation-induced strains. The Green-Lagrange strain is defined as:

$$[\varepsilon] = \frac{1}{2}\left([F]^T[F] - [I]\right) \tag{4.20}$$

In equation (4.20), $[F]$ is the deformation gradient and the product $[F][F]^T$ cancels out the effect of rigid body motion induced strains, while substracting the identity matrix gives an idea of how far is the current deformation state from the undeformed state. The key, then is to find the deformation gradient in curvilinear axis. The deformation gradient can be expressed as:

$$[F] = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \tag{4.21}$$

To do so, the outer product defined in subsection 4.1.3 comes in hand and is used to define the deformation gradient as:

$$[F] = \{g_i\} \otimes \{G^i\} = \begin{bmatrix} \frac{\partial x_1}{\partial x_i}\frac{\partial x_i}{\partial X_1} & \frac{\partial x_1}{\partial x_i}\frac{\partial x_i}{\partial X_2} & \frac{\partial x_1}{\partial x_i}\frac{\partial x_i}{\partial X_3} \\ \frac{\partial x_2}{\partial x_i}\frac{\partial x_i}{\partial X_1} & \frac{\partial x_2}{\partial x_i}\frac{\partial x_i}{\partial X_2} & \frac{\partial x_2}{\partial x_i}\frac{\partial x_i}{\partial X_3} \\ \frac{\partial x_3}{\partial x_i}\frac{\partial x_i}{\partial X_1} & \frac{\partial x_3}{\partial x_i}\frac{\partial x_i}{\partial X_2} & \frac{\partial x_3}{\partial x_i}\frac{\partial x_i}{\partial X_3} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \tag{4.22}$$

Once the deformation gradient has been calculated by the use of the covariants and contravariants the Right Cauchy-Green deformation tensor $[C]$ can be calculated as:

$$[C] = [F]^T[F] = \left(\{G^i\} \otimes \{g_i\}\right)\left(\{g_j\} \otimes \{G^j\}\right) = g_{ij}\{G^i\} \otimes \{G^j\} \tag{4.23}$$

where g_{ij} is the metric tensor of the covariant base. Expression (4.23) can be used to define the Green-Lagrange strain tensor. In order to obtain a simpler expression, the identity tensor $[I]$ in (4.20) is expressed in terms of the covariants base vectors in the reference configuration: $[G] = G_{ij}G^i \otimes G^j$. Thus, the strain tensor can be formulated as:

$$[\varepsilon] = \frac{1}{2}([C] - [G]) = \frac{1}{2}\left(g_{ij}\left(\{G^i\} \otimes \{G^j\}\right) - G_{ij}\left(\{G^i\} \otimes \{G^j\}\right)\right) \tag{4.24}$$
$$= \underbrace{\frac{1}{2}\left(g_{ij} - G_{ij}\right)}_{\text{covariant } \varepsilon_{ij}}\left(\{G^i\} \otimes \{G^j\}\right)$$

Equation (4.24) can be also seen from a geometrical point of view if expression (4.12) is seen as a measure of arc length. Then, what is done in (4.24) is basically to calculate the difference between two arc lengths. In addition, there is also a useful property implied in (4.24). The tensor $\varepsilon_{ij} = \frac{1}{2}(g_{ij} - G_{ij})$ is defined as the covariant strain tensor because it is defined by covariant quantities. Then it can be observed that the way to obtain the orthonormal components of a covariant tensor is to multiply it by the outer product of the contravariant base vectors.

Another important factor is that the treatment for the strain tensor seen above can be also applied to the stress tensor. Thus, the covariant or contravariant stress tensor may be defined in order to transform it back to the orthonormal base. The reciprocal base can be used in the same way as done for the strain tensor.

4.2.2 Constitutive relation

Once the strain tensor has been defined, a constitutive relation between the strain and stress tensor must be set. As the 4-node shell element must be able to model composite laminates, an orthotropic linear elastic constitutive matrix is defined through the compliance matrix $[\bar{C}]^{-1}$ where now the strain tensor is arranged as a vector: $\{\varepsilon\} = \{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23}\}^T$:

$$[\bar{C}]^{-1} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & 0 & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{23}} \end{bmatrix} \tag{4.25}$$

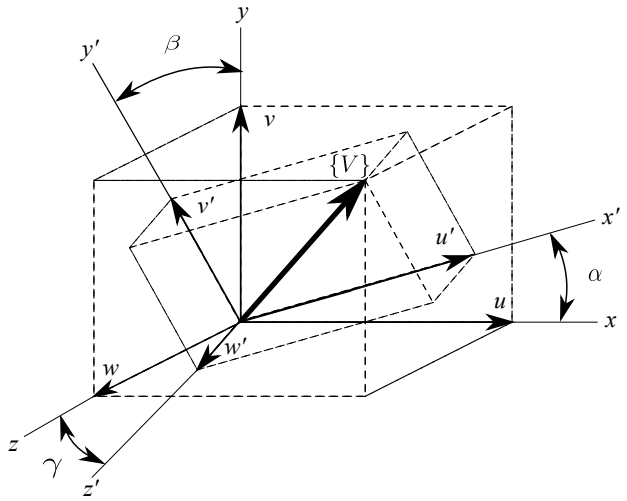
The constitutive relation fulfil the shell first shear deformation theory by having zeros in the terms associated with ε_3 due to the inextensional normal assumption. The bar in the constitutive matrix means that the constitutive matrix is set up in an orthonormal coordinate system aligned with the fibres so, a transformation that brings the constitutive relation to the covariant base is needed. In order to do so, the transformation matrix between non-orthonormal basis for the strain tensor needs to be described. In order to make it clearer for the reader and because it is also needed in the formulation of the element, the transformation needed for vectors is hereby presented.

In case of rotating nodal displacements and forces, the relation between two orthonormal coordinate systems, (such as the global Cartesian system and the element local Cartesian base)

is often called rotation matrix. Referred to as $[\Lambda]$, the rotation matrix is defined as follows

$$[\Lambda] = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} = \begin{bmatrix} cos(x', x) & cos(x', y) & cos(x', z) \\ cos(y', x) & cos(y', y) & cos(y', z) \\ cos(z', x) & cos(z', y) & cos(z', z) \end{bmatrix} \tag{4.26}$$

where l_i, m_i, n_i are the scalar products between the reference and destination axes also called direction cosines between the axes of the coordinate systems. Direction cosines matrices are used to express one set of orthonormal basis vectors in terms of another set, or for expressing a known vector in a different basis. Figure 4.5 serves as a visual aid for such concept.



	x	y	z
x'	l_1	m_1	n_1
y'	l_2	m_2	n_2
z'	l_3	m_3	n_3

Figure 4.5. Vector $\{V\}$ can be expressed in terms of components $\{u\}, \{v\}, \{w\}$ or $\{u'\}, \{v'\}, \{w'\}$. Direction cosines are angles between axes.

From figure 4.5 the mathematical manner to express vector $\{V\}$ in the new coordinate system (x', y', z') is achieved by means of $[\Lambda]$, also called rotation matrix.

$$\begin{Bmatrix} u' \\ v' \\ w' \end{Bmatrix} = [\Lambda] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{4.27}$$

When it comes to strain transformations the reader must be aware that they are transformations of displacement derivatives. As stated in Cook et al. (2001, Chap. 8), to relate strains in (x', y', z') coordinates to strains in (x, y, z) coordinates, displacement derivatives such as $\frac{\partial u'}{\partial x'}$ must be related to $\frac{\partial u}{\partial x}$ among other derivatives. In order to understand the general expression for the transformation a simple example is derived here. Let's relate the partial derivatives presented in the previous sentence.

$$u' = l_1 u + m_1 v + n_1 w \tag{4.28}$$

So to relate its derivatives we have

$$\frac{\partial u'}{\partial x'} = \frac{\partial u}{\partial x'} l_1 + \frac{\partial v}{\partial x'} m_1 + \frac{\partial w}{\partial x'} n_1 \tag{4.29}$$

Using the chain rule:

$$\frac{\partial u}{\partial x'} l_1 = \frac{\partial u}{\partial x} \frac{\partial x}{\partial x'} l_1 \tag{4.30}$$

Now, from (4.26) applied to the position vector we obtain that $\frac{\partial x}{\partial x'} = l_1$. Therefore,

$$\frac{\partial u'}{\partial x'} = \frac{\partial u}{\partial x} l_1^2 + \frac{\partial v}{\partial x} m_1^2 + \frac{\partial w}{\partial x} n_1^2 \tag{4.31}$$

Applying this process to all the displacement derivatives and connecting them properly so to obtain the appropriate strains, we obtain

$$\{\varepsilon'\} = [T]\{\varepsilon\} \tag{4.32}$$

where

$$[T] = \left[\begin{array}{ccc:ccc} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_1^3 & m_1^3 & n_1^3 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ \hdashline 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & l_1 m_2 + l_2 m_1 & m_1 n_2 + m_2 n_1 & n_1 l_2 + n_2 l_1 \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & l_2 m_3 + l_3 m_2 & m_2 n_3 + m_3 n_2 & n_2 l_3 + n_3 l_2 \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & l_3 m_1 + l_1 m_3 & m_3 n_1 + m_1 n_3 & n_3 l_1 + n_1 l_3 \end{array}\right] \tag{4.33}$$

The inverse of the matrix defined in (4.33) is its transposed. It is also known that the stored energy U is an invariant quantity that does not depend on the base you calculate express it. Therefore:

$$U = \frac{1}{2}\{\bar{\varepsilon}\}^T[\bar{C}]\{\bar{\varepsilon}\} = \frac{1}{2}\{\varepsilon\}^T[C]\{\varepsilon\} \tag{4.34}$$

where quantities with a bar are expressed in an orthonormal base while the non bar quantities are expressed in the covariant base. If equation (4.34) has to be fulfilled the following needs to hold according to (Cook et al., 2001, chap. 8).

$$[C] = [T]^T[\bar{C}][T] \tag{4.35}$$

Variational formulation 5

This chapter uses the concepts and derivations presented in chapter 4 to formulate the governing equations in which the finite element formulation is based for the 4-node enhanced shell element. The chapter is divided in two sections that formulate the boundary value problem formulation and then linearize the governing equation.

5.1 General variational formulation

In this section the variational formulation that was presented in Klinkel and Wagner (1997) and Klinkel et al. (1999), taking offset in Simo and Rifai (1990), is derived in more detail and remarking the steps that will have an effect on the finite element formulation later on. The original three field variational formulation was first introduced by Washizu (1975).

The formulation is based on the so called Enhanced Assumed Strains (EAS) method that is a type of incompatible enhancements. Basically, the strain field is enhanced by adding internal degrees of freedom. This is done by forcing the strain field to fulfill some requirements while deriving the variational formulation. Thus, the following strain field is assumed:

$$\{\varepsilon\} = \{\varepsilon^c\} + \{\hat{\varepsilon}\} \tag{5.1}$$

where $\{\hat{\varepsilon}\}$ is the incompatible strain enhancing term and the compatible or standard strain term is $\{\varepsilon^c\}$, and follows (4.24). The resulting strain from this addition $\{\varepsilon\}$ is therefore an enhanced strain.

In order to formulate the governing equation for the mechanical problem, the internal and external potential need to be defined to obtain the total potential energy.

$$\Pi(\{u\}, \{\hat{\varepsilon}\}, \{\hat{S}\}) = \Pi_i(\{u\}, \{\hat{\varepsilon}\}, \{\hat{S}\}) + \Pi_e(\{u\}) \tag{5.2}$$

In equation (5.2) the resulting potential depends on three fields instead of only one, as the enhancing strain $\{\hat{\varepsilon}\}$ and its corresponding enhancing stresses $\{\hat{S}\}$ are independent tensors introduced in this formulation. This means that they can be used and manipulated as desired in order to get simpler final formulation. If the material is assumed to be homogeneous and linear elastic, the internal potential can be expressed as:

$$\Pi_i = \int_{V_0} \Big(W_0(\{\varepsilon^c\} + \{\hat{\varepsilon}\}) - \{\hat{S}\} : \{\hat{\varepsilon}\} \Big)\, dV \tag{5.3}$$

W_0 is the function that gives the stored energy in the initial configuration. If the body is loaded by body forces $\{b\}$ and surface tractions $\{\Phi\}$, the external potential is expressed as:

$$\Pi_e = -\int_{V_0} \{b\} \cdot \{u\} dV - \int_{S_0} \{\Phi\} \cdot \{u\} dS \tag{5.4}$$

In order to get the governing equations, the first variation of the potential energy needs to be calculated and equated to zero. To do so the directional derivative proposed by Gâteaux is used. For a functional with one independent function is defined as:

$$\delta I(x;h) = \frac{d}{d\varepsilon} I(x+\varepsilon h)\Big|_{\varepsilon=0} \tag{5.5}$$

Expression (5.5) reads as the first variation of a functional that depends on x and it is varied with a quantity h. As explained in Dym and Shames (2013), the multiplier ε (do not confuse with the enhanced strain tensor) equating to zero is a condition for having a path (displacement and enhanced quantities fields) that extremizes the functional. In the case of having more than one independent function in the functional, the variation will be the summation of the variations of each function for the desired directions. Thus, the first variation of the total potential energy at the point $(\{u\},\{\hat{\varepsilon}\},\{\hat{S}\})$ in the direction $(\delta\{u\},\delta\{\hat{\varepsilon}\},\delta\{\hat{S}\}))$ is expressed as:

$$\begin{aligned} \delta\Pi = D\Pi(\{u\},\{\hat{\varepsilon}\},\{\hat{S}\})\cdot(\delta\{u\},\delta\{\hat{\varepsilon}\},\delta\{\hat{S}\}) \equiv \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\Pi(\{u_\varepsilon\},\{\hat{\varepsilon}_\varepsilon\},\{S_\varepsilon\}) \\ \{u_\varepsilon\} = \{u\} + \varepsilon\delta\{u\} \\ \{\hat{\varepsilon}_\varepsilon\} = \{\hat{\varepsilon}\} + \varepsilon\delta\{\hat{\varepsilon}\} \\ \{S_\varepsilon\} = \{\hat{S}\} + \varepsilon\delta\{\hat{S}\} \end{aligned} \tag{5.6}$$

The governing equation then can be expressed as:

$$\delta\Pi = \frac{\partial\Pi(\{u\}+\varepsilon\delta\{u\})}{\partial\varepsilon} + \frac{\partial\Pi(\{\hat{\varepsilon}\}+\varepsilon\delta\{\hat{\varepsilon}\})}{\partial\varepsilon} + \frac{\partial\Pi(\{\hat{S}\}+\varepsilon\delta\{\hat{S}\})}{\partial\varepsilon} = 0 \tag{5.7}$$

Calculating the derivatives, the following is obtained:

$$\begin{aligned} \delta\Pi = -\int_{V_0}\delta\{\hat{S}\}:\{\hat{\varepsilon}\} + \\ \int_{V_0}\left(\frac{\partial W_0}{\partial\{\varepsilon\}} - \{\hat{S}\}\right):\delta\{\hat{\varepsilon}\}dV + \\ \int_{V_0}\frac{\partial W_0}{\partial\{\varepsilon\}}:\delta\{\varepsilon^c\}dV - \int_{V_0}\{\Phi\}\delta\{u\}dV - \int_S\{b\}\delta\{u\}dS = 0 \end{aligned} \tag{5.8}$$

The first variation of the compatible strain tensor $\{\varepsilon^c\}$ (that depends on the displacements) can be derived following the same approach as in (5.6):

$$\delta\{\varepsilon^c\} = \frac{d}{d\varepsilon}\{\varepsilon^c(\{g_i\}+\varepsilon\delta\{g_i\})\}\Big|_{\varepsilon=0} \tag{5.9}$$

Substituting terms in expression (4.24) and using the definition of the metric tensor given in (4.11):

$$\delta\{\varepsilon^c\} = \frac{d}{d\varepsilon}\frac{1}{2}\left((\{g_i\}+\varepsilon\delta\{g_i\})\cdot(\{g_j\}+\varepsilon\delta\{g_j\}) - \{G_i\}\cdot\{G_j\}\right)\{G^i\}\otimes\{G^j\}\Big|_{\varepsilon=0} \tag{5.10}$$

Notice that in expression (5.10) the covariants and the contravariants expressed in the reference configuration are not affected by the displacements and therefore, the variation is not applied to them. If the derivative is calculated the following is obtained:

$$\frac{1}{2}\left(\delta\{g_i\}\cdot\{g_j\} + \delta\{g_i\}\varepsilon\{g_j\} + \{g_i\}\delta\{g_j\} + \varepsilon\delta\{g_j\} + \varepsilon\delta\{g_i\}\delta\{g_i\}\right)\{G^i\}\otimes\{G^j\}\Big|_{\varepsilon=0} \tag{5.11}$$

Evaluating expression (5.11) the first variation of the compatible strain tensor is:

$$\delta\{\varepsilon\} = \frac{1}{2}\left(\delta\{g_i\}\cdot\{g_j\} + \{g_i\}\delta\{g_j\}\right)\{G^i\}\otimes\{G^j\} \tag{5.12}$$

5.2 Linearization of the governing equation

Equation (5.8) represents the governing equation that can be solved by an iterative method such as the Newton-Raphson algorithms family. In order to understand why and how the linearization is done the Newton-Raphson algorithm ideas are introduced in the following lines.

5.2.1 General Newton-Raphson method

The Newton-Raphson method is an iterative solution to the problem of finding the zero of a function. Let the function be the residual vector obtained from setting equilibrium between externally applied forces $\{f_{ext}\}$ and internal forces $\{f_{int}\}$. Thus, the residual is defined as $\{R(\{u\},\{f_{ext}\})\} = \{f_{int}(u)\} - \{f_{ext}\}$ and its solution equating to zero is sought.

The iterative manner proposed by the Newton-Raphson method starts by setting a first order Taylor expansion at a chosen point $\{u_i\}$.

$$\{R(\{u_{k+1}\},\{f_{ext}\})\} = \{R(\{u_k\},\{f_{ext}\})\} + \frac{\partial\{R(\{u_k\},\{f_{ext}\})\}}{\partial\{u\}}(\{u_{k+1}\} - \{u_k\}) \tag{5.13}$$

where the higher terms are ignored.

The solution $\{R(\{u\},\{f_{ext}\})\} = 0$ is sought and by introducing the increment $\{\Delta u\} = \{u_{k+1}\} - \{u_k\}$ we obtain

$$\{R(\{u_k\},\{f_{ext}\})\} + \frac{\partial\{R(\{u_k\},\{f_{ext}\})\}}{\partial\{u\}}\{\Delta u\} = \{0\} \tag{5.14}$$

and thereby

$$\frac{\partial\{R(\{u_k\},\{f_{ext}\})\}}{\partial\{u\}}\{\Delta u\} = -\{R(\{u_k\},\{f_{ext}\})\} \tag{5.15}$$

If this approximation is repeated in an iterative process updating the displacement with the result from the incremental equilibrium in (5.15)

$$\{u_{k+1}\} = \{u_k\} + \{\Delta u\}$$

a precise result depending on the number of iterations can be found.

From these basic formulation there are different ramifications that work better for different problems. At the end though, the problem is always reduced to being able to compute the residual in order to find an accurate result, and the first derivative in order to find it with less iterations.

5.2.2 Linearised governing equation

It has been seen in section (5.2.1) that in order to apply a Newton-Raphson based algorithm the computation of the first derivative of the function being solved is needed. In the case that occupies this work ($\delta\Pi = 0$) the approximation of the potential energy first variation can be expressed as:

$$\begin{aligned}&\delta\Pi(\{u_{k+1}\},\{\hat{\varepsilon}_{k+1}\},\{\hat{S}_{k+1}\}) = \\ &\delta\Pi(\{u_k\},\{\hat{\varepsilon}_k\},\{\hat{S}_k\}) + D\delta\Pi(\{u_k\},\{\hat{\varepsilon}_k\},\{\hat{S}_k\})\cdot(\Delta\{u_{k+1}\},\Delta\{\hat{\varepsilon}_{k+1}\},\Delta\{\hat{S}_{k+1}\}) = \{0\}\end{aligned} \tag{5.16}$$

It can be observed that the second right hand term is actually really similar to the one described in (5.6). Actually, this term is the directional derivative of the first variation of the potential,

which is its second variation ($\Delta\delta\Pi$). If this is calculated with the same method used in (5.6), the following is reached:

$$\begin{aligned}\Delta\delta\Pi = &\int_{V_0} \delta\{\varepsilon^c\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\varepsilon^c\} + \frac{\partial W_0}{\partial\{\varepsilon\}} : \delta\Delta\{\varepsilon^c\} dV + \int_{V_0} \delta\{\varepsilon^c\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\hat{\varepsilon}\} dV \\ &+ \int_{V_0} \delta\{\hat{\varepsilon}\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\varepsilon^c\} dV + \int_{V_0} \delta\{\hat{\varepsilon}\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\hat{\varepsilon}\} dV - \int_{V_0} \delta\hat{S} : \Delta\{\hat{\varepsilon}\} dV \\ &- \int_{V_0} \delta\{\hat{\varepsilon}\} : \Delta\{\hat{S}\} dV \end{aligned} \tag{5.17}$$

The compatible strain tensor appears as a linearised quantity in (5.17). Following the calculation done in the previous section for the strain tensor, (5.9) to (5.12), the linearised strain tensor is expressed as:

$$\Delta\delta\{\varepsilon^c\} = \frac{1}{2} \left(\Delta\{g_i\} \cdot \delta\{g_j\} + \delta\{g_i\} \cdot \Delta\{g_j\}\right) \{G^i\} \otimes \{G^j\} \tag{5.18}$$

In order to be able to solve the problem by only having to enhance the strains, a condition on the enhancing strain field $\{\hat{\varepsilon}\}$ is imposed. This can be done because the enhancing strain field is independent and artificially introduced. Specifically, the enhancing strain field needs to be orthogonal with the stress field associated to it. If this is assured the following holds because of the properties of the colon product:

$$\int_{V_0} \{\hat{S}\} : \{\hat{\varepsilon}\} dV = 0 \tag{5.19}$$

Thus, expression (5.17) becomes a two independent field variation:

$$\begin{aligned}\Delta\delta\Pi = &\int_{V_0} \delta\{\varepsilon^c\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\varepsilon^c\} + \frac{\partial W_0}{\partial\{\varepsilon\}} : \delta\Delta\{\varepsilon^c\} dV + \int_{V_0} \delta\{\varepsilon^c\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\hat{\varepsilon}\} dV \\ &+ \int_{V_0} \delta\{\hat{\varepsilon}\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\varepsilon^c\} dV + \int_{V_0} \delta\{\hat{\varepsilon}\} : \frac{\partial^2 W_0}{\partial\{\varepsilon\}^2} : \Delta\{\hat{\varepsilon}\} dV \end{aligned} \tag{5.20}$$

Equation (5.20) completes the formulation of the variational problem. From this point, the expression on (5.16) needs to be equalised to zero to find the equilibrium state. However, as this is not possible analytically, the finite element method needs to be introduced to discretise the two variational fields and solve the problem.

Finite element formulation 6

This chapter presents the finite element formulation applied in order to solve the governing equations derived in chapter 5. The goal of this section is to understand where the different stiffness matrices needed in the finite element analysis for the 4-node enhanced shell element originate using the variational formulation from the previous chapter. The numerical implementation of the stiffness matrices is also explained, as well as the linear buckling formulation.

6.1 Discretisation of the shell domain

The Reissner-Mindlin assumptions can be used to formulate the 4-node shell element through the degeneration process, which is the most common way to formulate the kinematics of shell elements because it is easy and computationally cheap. The formulation described in this section follows the one described in Lund (2016a) or Bathe (1996).

The degeneration process is based on the isoparametric mapping explained in appendix A that uses the following shape functions for a 3-D trilinear brick element:

$$N_I(r,s,t) = \frac{1}{8}(1 \pm r)(1 \pm s)(1 \pm t) \tag{6.1}$$

In (6.1) N_I is the shape function associated to the node I that maps the element from the natural space (r,s,t) to the orthonormal space (x,y,z). The position and displacement in the global space can be, according to Cook et al. (2001), expressed as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} \sum_{I=1}^{n} N_I x_I \\ \sum_{I=1}^{n} N_I y_I \\ \sum_{I=1}^{n} N_I z_I \end{Bmatrix} \qquad \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} \sum_{I=1}^{n} N_I u_I \\ \sum_{I=1}^{n} N_I v_I \\ \sum_{I=1}^{n} N_I w_I \end{Bmatrix} \tag{6.2}$$

The node directors are calculated as suggested in Bathe (1996) and can be seen in figure 6.1.

$$\{V_{10}^I\} = \frac{\{j\} \times \{V_{30}^I\}}{|\{j\} \times \{V_{30}^I\}|}, \qquad \{V_{20}^I\} = \{V_{30}^I\} \times \{V_{10}^I\}, \qquad \{V_{30}^I\} = \frac{\{X_{top}^I\} - \{X_{bottom}^I\}}{h_I} \tag{6.3}$$

They are a set of vectors conforming an orthonormal coordinate system attached to each node of the element. Their main purpose is to define the thickness direction by means of the normal vector $\{V_{30}^I\}$. This vector allows extrapolating any field value through-the-thickness from the shell middle surface.

Next, the shape functions are separated in two expressions: one linked to the in-plane interpolation (N_I^{2D}) and another linked to the thickness interpolation (N_I^t).

$$N_I^{2D} = \frac{1}{4}(1 \pm r)(1 \pm s), \qquad N_I^t = \frac{1}{2}(1 \pm t) \tag{6.4}$$

Thus, the geometrical interpolation can be rewritten as

$$\{x\} = \sum_{I=1}^{n} \frac{1}{2}(1+t)N_I^{2D}\{x_{top}^I\} + \sum_{I=1}^{n} \frac{1}{2}(1-t)N_I^{2D}\{x_{bottom}^I\} \tag{6.5}$$

Here n is now 4 as the element has been degenerated to a 4-node shell. A rearrangement is needed in order to be able to obtain a more compact expression:

$$\{x\} = \sum_{I=1}^{4} N_I^{2D}\frac{1}{2}\Big(\{x_{top}^I\} + \{x_{bottom}^I\}\Big) + \sum_{I=1}^{4} \frac{t}{2}N_I^{2D}(\{x_{top}^I\} - \{x_{bottom}^I\}) \tag{6.6}$$

Using the node director definition $\{V_{30}^I\} = \frac{\{X_{top}^I\}-\{X_{bottom}^I\}}{h_I}$, any point at the line between the two former nodes of the solid element can be expressed as the midsurface node coordinates plus the node director multiplied by a thickness value. Using this property, the expression in (6.6) can be written as

$$\{x\} = \sum_{I=1}^{4} N_I^{2D}\{x_I\} + \sum_{I=1}^{4} \frac{t}{2}N_I^{2D}h_I\{V_{30}^I\} \tag{6.7}$$

Equation (6.7) depends only on the mid-surface node positions $\{x_I\}$ and the associated node directors.

Once the interpolation of the position has been derived, the last step in order to obtain the kinematics of the degenerated element is to interpolate the nodal displacements. The displacement can be expressed as the difference between the position of the point after and before deformation. With this in mind and making use of the node directors again:

$$\{u\} = \sum_{I=1}^{4} N_I(\{x_t^I\}-\{x_0^I\}) + \sum_{I=1}^{4} \frac{t}{2}N_I h_I(\{V_{3t}^I\}-\{V_{30}^I\}) = \sum_{I=1}^{4} N_I\{u^I\} + \sum_{I=1}^{4} \frac{t}{2}N_I h_I(\{V_{3t}^I\}-\{V_{30}^I\}) \tag{6.8}$$

The next step is expressing the difference between the node directors in (6.8) in a more "useful" way for the analysis. As done in Lund (2016a), the node directors $\{V_{20}^I\}$ and $\{V_{10}^I\}$ are introduced. The objective of doing so is to be able to express the difference by the use of node directors in the undeformed state and two rotations that will constitute the two rotational degrees of freedom. Figure 6.1 shows how the node directors are defined and how the rotations are used.

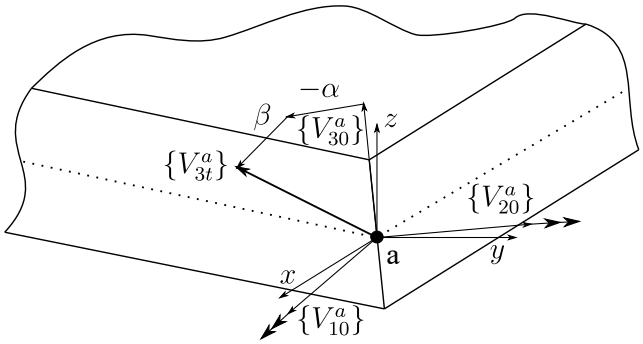


Figure 6.1. Definition of the node directors and visualization of the rotational degrees of freedom.

If a rotation α is performed around $\{V_{10}^I\}$ node director, and because small rotations are assumed, a displacement α in the negative direction of $\{V_{20}^I\}$ appears. In the same way, if a β

rotation around $\{V_{20}^I\}$ is performed, a displacement β appears in the $\{V_{10}^I\}$ direction. With these definitions, the displacement can be expressed as

$$\{u\} = \sum_{I=1}^{4} N_I\{u_I\} + \sum_{I=1}^{4} \frac{t}{2} h_I N_I(-\alpha_I\{V_{20}^I\} + \beta_I\{V_{10}^I\}) \tag{6.9}$$

Thus, the interpolation matrix can be expressed as:

$$[N] = \left[\begin{array}{c|cccc|c} & N_I & 0 & 0 & -\frac{t}{2}N_I h_I\{V_{21}^I\} & \frac{t}{2}N_I h_I\{V_{11}^I\} & \\ \dots & 0 & N_I & 0 & -\frac{t}{2}N_I h_I\{V_{22}^I\} & \frac{t}{2}N_I h_I\{V_{12}^I\} & \dots \\ & 0 & 0 & N_I & -\frac{t}{2}N_I h_I\{V_{23}^I\} & \frac{t}{2}N_I h_I\{V_{13}^I\} & \end{array}\right] \tag{6.10}$$

6.2 Coordinate systems and bases

On the formulation of the 4-node shell element, different coordinate systems and space bases are needed. They are used to properly define the features that characterize the 4-node shell element and its enhancements. The relation within coordinate systems varies depending on which field is being transformed. When dealing with positions or displacements they can be graphically represented by a vector. Therefore, their expression in another coordinate system requires the use of a rotation matrix. If we seek the relation between the strain or stress quantities in one coordinate system and the same quantity is expressed in another coordinate system, the transformation becomes more complex. In order to formulate the shell element of this work the following coordinate systems and bases are needed:

- Cartesian 3D coordinate system. Global coordinate system (x, y, z)
- Curvilinear coordinate system. Natural element mapping (r, s, t)
- Global orthonormal standard basis $(\{e_1\}, \{e_2\}, \{e_3\})$
- Local orthonormal basis at the element center $(\{d_1\}, \{d_2\}, \{d_3\})$
- Covariant basis at a defined point in the reference state $(\{G_1\}, \{G_2\}, \{G_3\})$ and in the current state $(\{g_1\}, \{g_2\}, \{g_3\})$

Each of the previously listed coordinate systems has some properties that make them being defined in a specific manner and gives them useful features to be used while performing calculations on the finite element. These are described in the following subsections.

6.2.1 Coordinate systems

Cartesian 3D coordinate system

As stated in Smith and Latham (1925) a Cartesian 3D coordinate system is a coordinate system that specifies each point uniquely as the signed distances to each of the three perpendicular planes. The intersection of these planes are called coordinate axes and they meet at the origin which corresponds to $(0, 0, 0)$ in a $\mathbb{R}^3$ mapping. The coordinates can also be defined as the positions of the perpendicular projections of the point onto the three axes, expressed as signed distances from the origin.

The Cartesian coordinate system is used to algebraically express Euclidean geometry, and in the case of this work, it serves as a Global reference for all the points of the mesh belonging to the input geometry to be analyzed.

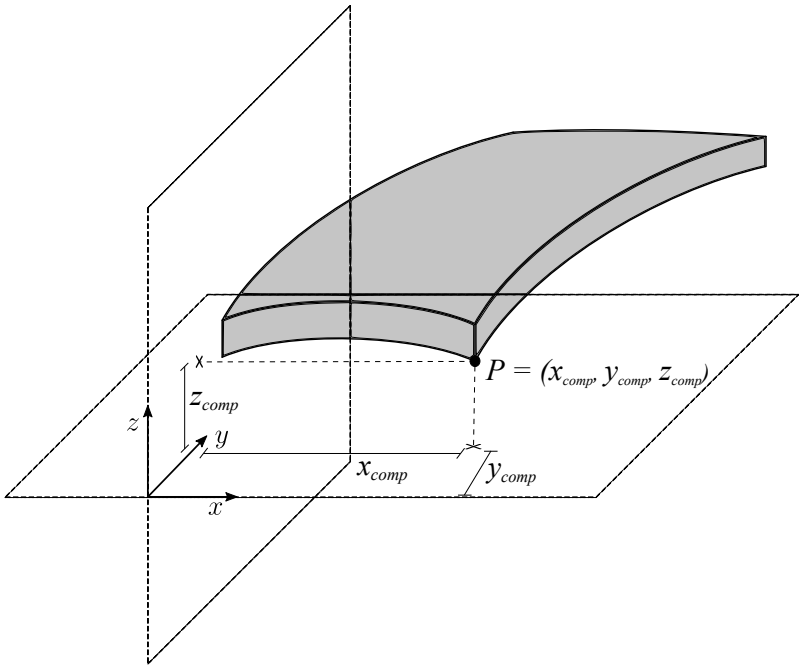


Figure 6.2. Global Cartesian coordinate system (x, y, z).

Curvilinear coordinate system

Curvilinear coordinates are used to define a Euclidean space that may be curved. They adapt to the physical surface to be described. For the double curved shell element of this work, such reference system is defined by attaching the axes of coordinates to the middle point of both element edges. The transversal axis is defined as normal to the surface in the element center (see figure 6.3).

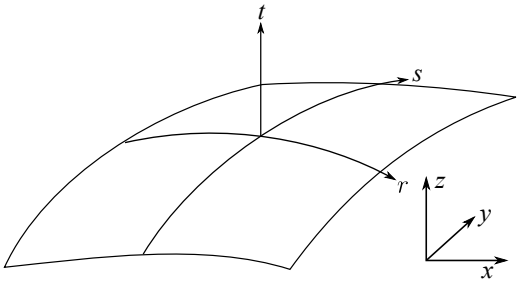


Figure 6.3. General shell with the global coordinates (x, y, z) and the convective curvilinear coordinates (r, s, t).

As explained in appendix A the isoparametric formulation makes the hexahedral element with non-rectangular shape compatible. To do so, it requires the convective coordinate system (r, s, t) which for the 3D shell element of this work, maps the physical element into a quadrilateral with edges of length 2.

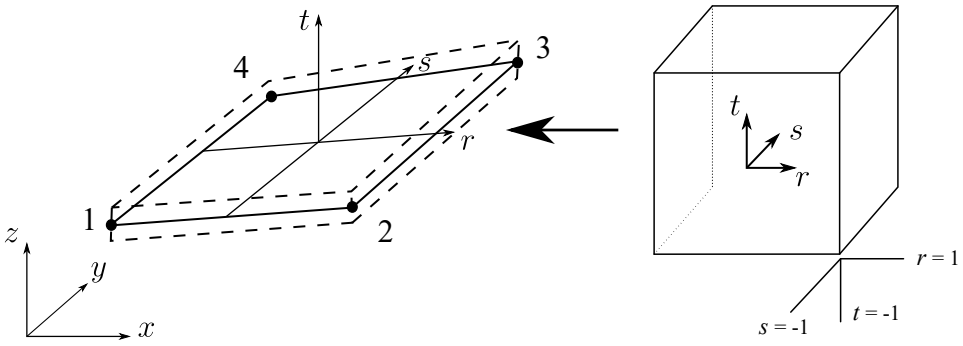


Figure 6.4. Isoparametric mapping.

When located in the local, natural, convective element coordinate system, any point of the geometry of the element is described by coordinates ranging from $[-1, 1]$ in the r, s or t directions. By means of the chain rule of differentiation and the element interpolation functions, any geometry or displacement value can be computed in the element. The coordinates are specified as (r, s, t) values from $[-1, 1]$, interpolated along the element with the influence of nodal values and then transformed to another basis with the Jacobian matrix of the transformation, which is used to transform a Cartesian basis to a curvilinear one, and vice versa.

6.2.2 Bases

Global standard basis

The standard basis for a Euclidean space is the set of unit vectors pointing in the direction of the axes of a Cartesian coordinate system. The 4-node shell element with MITC and EAS enhancements is formulated to work in the three-dimensional space, therefore, the standard basis is defined as follows

$$\{e_1\} = (1, 0, 0) \qquad \{e_2\} = (0, 1, 0) \qquad \{e_3\} = (0, 0, 1) \tag{6.11}$$

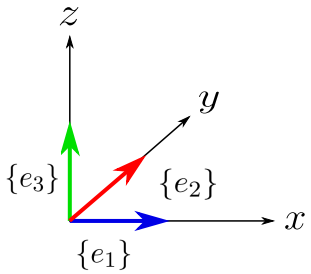


Figure 6.5. Cartesian reference system (x, y, z) and the standard basis $(\{e_1\}, \{e_2\}, \{e_3\})$.

Equation (6.11) defines a basis because the vectors are linearly independent in the sense that any other vector can be expressed as a unique combination of these three. It is global because their vectors are the same at all points of the geometry (Boothby, 1975).

Covariant basis at a defined point

The theory of the covariance and the covariant basis is introduced and defined in section 4.1. Equation (4.14) describes the computation of the covariant base vectors. They are defined as the

derivatives of the reference position vector $\{X\}$ with respect to the natural element coordinates (r, s, t), repeated here for convenience:

$$\{G_i\} = \frac{\partial\{X\}}{\partial r_i} = \{\frac{\partial X}{\partial r_i}, \frac{\partial Y}{\partial r_i}, \frac{\partial Z}{\partial r_i}\}\{e_k\} \ , \quad r_i = r, s, t \quad \text{being} \quad i, k = 1, 2, 3 \tag{6.12}$$

The covariant base at any point is always tangent to the shell surface. In addition, it also scales the isoparametric mapping by means of the Jacobian while changing coordinate systems. For these reasons, it is used in this work as the vector basis in which to define the element strain-displacement matrix $[B]$ as well as to define the strain enhancements for the formulation.

As explained in section 4.2, the covariant basis is used to simplify the handling of vector components under non-orthonormal bases.

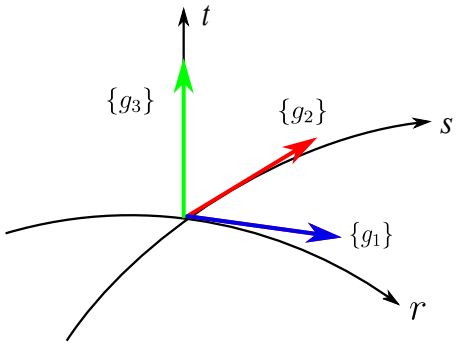


Figure 6.6. Curvilinear, convective coordinates (r, s, t) and its covariant base $(\{g_1\}, \{g_2\}, \{g_3\})$ in the current configuration.

As mentioned at the beginning of the section, the covariant base at a defined point in the current configuration is also needed in this work. The calculation of such covariant base is done as in (6.12) but changing the reference position vector $\{X\}$ by the current configuration position vector $\{x\}$. The resulting base is expressed as: $(\{g_1\}, \{g_2\}, \{g_3\})$.

Local orthonormal basis

The local element basis is computed with the help of the covariant base of the element in the center of it's geometry. As explained previously, the covariant base is directly related to the position, therefore such element basis is local, meaning that it changes for each element. It is a Cartesian base defined as similar as possible to the covariant base but assuring orthonormality of its vectors. Such system can be defined as (Hughes, 1987)

$$\{d_1\} = \frac{\sqrt{2}}{2}(a - b) \qquad \{d_2\} = \frac{\sqrt{2}}{2}(a + b) \qquad \{d_3\} = \text{normal to the surface} \tag{6.13}$$

where

$$\{a\} = \frac{\frac{1}{2}(\{g_1\} + \{g_2\})}{\|\frac{1}{2}(\{g_1\} + \{g_2\})\|} \qquad \{b\} = \frac{\{d_3\} \times \{a\}}{\|\{d_3\} \times \{a\}\|}$$

As the formulation of the 4-node shell element is made for laminated composite structures, an orthotropic constitutive relation needs to be properly defined. The matrix of material constants $[C]$ is only well defined in a Cartesian base (Lund, 2016a). Thus, the local element coordinate system $(\{d_1\}, \{d_2\}, \{d_3\})$ provides a local Cartesian base close to the average middle plane of the double curved element, where the constitutive matrix can be defined.

The above presented local base is a first approximation taken in the definition of the 4-node linear flat shell element with EAS and MITC enhancements developed in the previous semester. The non-linear 3D implementation in MUST and its linearized version compute the element local base by using projections of the element coordinate system defined in the ANSYS mesh (ESYS). This new implementation makes the laminate definition more comfortable because it is easier to orient them on such a defined base.

To conclude, the element local base can be defined in different ways and if the material is isotropic the definition is rather irrelevant. However, if the element needs to be able to represent laminates of orthotropic materials a base that facilitates the task of defining the laminates is very useful.

6.3 Geometrically non-linear analysis

Once the discretization has been done and the bases that are used within the element formulation have been defined, the variational formulation needs to be transformed to a finite element form. The goal of this section is to end up having an expression of the tangent stiffness $[K_T]$ that can be used to formulate the incremental displacement equation $[K_T]\Delta\{D\} = -\{R\}$ in which the residual $\{R\}$ is defined as $\{R\} = \{F_{int}\} - \{F_{ext}\}$ (section 5.2.1).

Notice that the tangent stiffness is related to the variational formula (5.20), while the residual part is related to the variation (5.6).

There are two interpolation matrices needed in the formulation presented in this work, one that links the displacements with the compatible strain field $\{\tilde{\varepsilon}^c\}$ in the covariant base, and one that links the internal degrees of freedom with the enhancing strain field $\{\hat{\varepsilon}\}$. From now on everything is formulated at an element level to be assembled later on to a global level.

6.3.1 Non-linear strain-displacement matrix

The strain-displacement matrix $[B_I]$ relates the first variation of the compatible strain field with the first variation of the displacement field:

$$\delta\{\tilde{\varepsilon}^c\} = \sum_{I=1}^{4} [B_I]\delta\{d_I\} \tag{6.14}$$

The matrix $[B_I]$ is derived taking into account (5.12) that gives the relation between the covariants and the strain tensor. According to (6.14) the following expression needs to be obtained:

$$\begin{pmatrix} \delta\tilde{\varepsilon}_{11} \\ \delta\tilde{\varepsilon}_{22} \\ 0 \\ \delta\tilde{\gamma}_{12} \\ \delta\tilde{\gamma}_{23} \\ \delta\tilde{\gamma}_{13} \end{pmatrix}^c = \sum_{I=1}^{4} [B_I] \begin{pmatrix} \delta u_I \\ \delta v_I \\ \delta w_I \\ \delta \alpha_I \\ \delta \beta_I \end{pmatrix} \tag{6.15}$$

It is clear then, that the $[B_I]$ matrix needs to be of dimension 6×5 for the node I. An added problem that the non-linear formulation induces is that the strain-displacement matrix is depending on the covariant vectors in the current configuration.

In order to be able to compute the $[B_I]$ matrix, the covariant vectors in the current state are split in two parts:

$$\{g_i\} = \frac{\partial\{x\}}{\partial x_i} = \frac{\partial(\{X\} + \{d\})}{\partial x_i} = \frac{\partial\{X\}}{\partial x_i} + \frac{\partial\{u\}}{\partial x_i} \tag{6.16}$$

The first part can be evaluated from the reference configuration while the second one can be evaluated at every iteration of the solution algorithm for the current deformed state, leading to the following expression:

$$[B_I] = [B_0(\{G_i\})] + [B_{NL}(\{d\}] \tag{6.17}$$

The advantage of this transformation is that the code necessary in order to compute the part of the $[B_I]$ matrix that depends on the covariant vectors and the one depending on the displacement derivatives is the same (section 6.5 equation (6.54)). The only difference is that for the first case the input is the positions of the nodes in the original configuration, while for the second one, the input is the current displacement field. Precisely, if this separation of the covariant base vectors is introduced in the covariant part of (4.24):

$$\tilde{\varepsilon}_{ij} = \frac{1}{2}\left(\left(\frac{\partial(\{X\} + \{u\})}{\partial x_i}\frac{\partial(\{X\} + \{u\})}{\partial x_j}\right) - \left(\frac{\partial\{X\}}{\partial x_i}\frac{\partial\{X\}}{\partial x_j}\right)\right) \tag{6.18}$$

If the multiplication is done, the following result is obtained:

$$\tilde{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial\{X\}}{\partial x_i}\frac{\partial\{X\}}{\partial x_j} + \frac{\partial\{X\}}{\partial x_i}\frac{\partial\{u\}}{\partial x_j} + \frac{\partial\{u\}}{\partial x_i}\frac{\partial\{X\}}{\partial x_j} + \frac{\partial\{u\}}{\partial x_i}\frac{\partial\{u\}}{\partial x_j} - \frac{\partial\{X\}}{\partial x_i}\frac{\partial\{X\}}{\partial x_j}\right) \tag{6.19}$$

By using the definition of the covariant base:

$$\tilde{\varepsilon}_{ij} = \frac{1}{2}\left(\{G_i\}^T \cdot \frac{\partial\{u\}}{\partial x_j} + \{G_j\}^T \cdot \frac{\partial\{u\}}{\partial x_i} + \frac{\partial\{u\}}{\partial x_i}\frac{\{u\}}{\partial x_j}\right) \tag{6.20}$$

where the non-linear part of the equation is redefined as $\eta_{ij} = \frac{1}{2}\frac{\partial\{u\}}{\partial x_i} \cdot \frac{\partial\{u\}}{\partial x_j}$.

6.3.2 Enhanced assumed strains

The second relation matrix needed is the one relating the enhanced strain field $\{\hat{\varepsilon}\}$ with the internal degrees of freedom $\{\alpha\}$:

$$\{\hat{\varepsilon}\} = [\tilde{M}(r,s)]\{\alpha\} \tag{6.21}$$

The formulation of the enhanced assumed strains is directly related to the method of incompatible displacement modes developed in Wilson et al. (1973). If the reader is interested in more information about the link between these two formulations, Simo and Rifai (1990) offers a good explanation on the matter.

The incompatible strains $\{\hat{\varepsilon}\}$ are interpolated from the strains evaluated in the centre of the element in the natural coordinate system (r, s, t). The choice of this interpolation has two main reasons. In first place, calculating the strains in the isoparametric space allows to treat always with a quadrilateral element making the interpolation choice problem independent. In the second place, the centre of the element is chosen as it represents an average of the element distortion.

The relation between the enhanced strain in the isoparametric space at the centre of the element and the strain expressed in the covariant base at any point of the element is:

$$\{\tilde{\varepsilon}\} = \frac{\det[J_0]}{det[J]}[T_0]\{\hat{\varepsilon}\} \tag{6.22}$$

In (6.22) $\{\hat{\varepsilon}\}$ is the enhancing strain obtained from the interpolation shown in (6.21), the Jacobian matrix $[J]$ is expressed as $[J] = [\{G_1\}, \{G_2\}, \{G_3\}]$ and $[J_0]$ follows the same expression but with the covariants evaluated in the centre. $[T_0]$ is a transformation matrix defined as in (4.33) but having the direction cosines between $\{G_i\}$ and $\{G^j\}$ because in order to transform from the isoparametric space, the inverse of the Jacobian is needed. Due to the definition of the Jacobian (4.17), its inverse can be written in terms of the contravariants.

The enhancing strains in the centre of the element are calculated as in (6.22), where $\{\alpha\}$ represents the internal degrees of freedom and can be as large as the problem requires.

In order to properly define the $[M]$ matrix, the orthogonality condition expressed in (5.19) is used. Applying the orthogonality condition to (6.22):

$$\int_{V_E} \delta\{\hat{S}\}\frac{det[J_0]}{det[J]}[T_0]\{\hat{\varepsilon}\}det[J]drdsdt = 0 \tag{6.23}$$

The constant values can be disregarded giving the following expression:

$$\int_{V_E} [M(r,s)]drdsdt = 0 \tag{6.24}$$

From here, several interpolation matrices formulated through numerical experimentation can be chosen. In Klinkel et al. (1999) there are examples of different interpolation matrices but the one used for this formulation is inspired by Crisfield (2000, chap. 18.11.2) and shows accurate enough results with only four internal degrees of freedom:

$$[M] = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & s \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.25}$$

This matrix will have to be tested by a patch test to make sure the formulation of the element is correct. It can also be noted that the only components enhanced are $\varepsilon_{11}, \varepsilon_{22}$ and ε_{12} that are the in-plane components.

Eventually, the matrix that relates $\{\alpha\}$ and the covariant strain field $\{\tilde{\varepsilon}\}$ in (6.18) can be defined as:

$$[\tilde{M}(r,s)] = \frac{det[J_0]}{det[J]}[T_0][M(r,s)] \tag{6.26}$$

6.3.3 Governing equation in finite element form

In the following, the two relations derived in the previous section and the constitutive matrix derived in subsection 4.2.2 are used to reformulate the governing equation in a discretized form.

Because of the potential meaning of W_0, the 2nd Piola-Kirchoff stresses are defined as:

$$\{S\} = \frac{\partial W_0}{\partial \{\varepsilon\}} \tag{6.27}$$

That implies the stiffness matrix $[C]$ can be defined as:

$$[C] = \frac{\partial^2 W_0}{\partial \{\varepsilon\}^2} \tag{6.28}$$

Substituting in (5.20) and (5.6) and adding the results, the incremental form of the total potential is obtained.

$$\begin{aligned} &\int_{V_0} \{S\}\delta\{\hat{\varepsilon}\}dV + \int_{V_0} \{S\}\delta\{E^c\}dV - \int_{V_0} \{b\}\delta\{u\}dV \\ -\int_{S_0} \{\Phi\}\delta\{u\}dS + &\int_{V_0} \left(\delta\{\varepsilon^c\}[C]\Delta\{\varepsilon^c\} + \{S\}\delta\Delta\{\varepsilon^c\}\right)dV + \int_{V_0} \delta\{\varepsilon^c\}[C]\Delta\{\hat{\varepsilon}\}dV \\ &+ \int_{V_0} \delta\{\hat{\varepsilon}\}[C]\Delta\{\varepsilon^c\}dV + \int_{V_0} \delta\{\hat{\varepsilon}\}[C]\Delta\{\hat{\varepsilon}\}dV = 0 \end{aligned} \tag{6.29}$$

Now, the $[B_I]$ and $[\tilde{M}]$ matrices and the interpolation matrix $[N_I]$ in (6.10) are used to express (6.29) as:

$$\begin{aligned} &\sum_{I=1}^{4} \delta\{d_I\}^T \Bigg(\int_{V_0} [B_I]^T[S]dV - \int_{V_0} [N_I]^T\{b\}dV - \int_{S_0} [N_I]^T\{\Phi\}dS \\ &+ \sum_{J=1}^{4} \int_{V_0} ([B_I]^T[C][B_J] + [G_{IJ}])dV\Delta\{d_J\} + \int_{V_0} [B_I]^T[C][\tilde{M}]dV\delta\{\alpha\} \Bigg) \end{aligned} \tag{6.30}$$

$$+\delta\{\alpha\}^T \Bigg(\int_{V_0} [\tilde{M}]^T[S]dV + \sum_{J=1}^{4} \int_{V_0} [\tilde{M}]^T[C][B_J]dV\Delta\{d_J\} + \int_{V_0} [\tilde{M}]^T[C][\tilde{M}]dV\Delta\{\alpha\} \Bigg) = 0$$

From (6.30) and (6.29) the following stiffness matrices and load terms can be identified. Notice that the computation of the stiffness matrices linked to the compatible DOF $\{d_I\}$ are defined as a double summation over $I = J =$ number of nodes. On the other hand, the matrix components connected to the incompatible parameters $\{\alpha\}$ are computed element wise because their values are not nodal dependent.

Stiffness matrices:

$$[k_{IJ}] = \int_{V_e} \Big([B_I(\{d_I\})]^T[C][B_J(\{d_J\})] + [G_{IJ}(\{d_I\},\{d_J\})] \Big) dV \tag{6.31}$$

$$[L_I] = \int_{V_e} [\tilde{M}]^T[C][B_I(\{d_I\})]dV \tag{6.32}$$

$$[H_e] = \int_{V_e} [\tilde{M}]^T[C][\tilde{M}]dV \tag{6.33}$$

Load terms:

$$\{f_I^{int}\} = \int_{V_e} [B_I(\{d_I\}]^T[S(\{d_I\},\{\alpha_I\})]dV \tag{6.34}$$

$$\{f_I^{ext}\} = \int_{V_e} [N_I]^T\{b_I\}dV + \int_{S_e} [N_I]^T\{\Phi_I\}dS \tag{6.35}$$

$$\{h_e\} = \int_{V_e} [\tilde{M}]^T[S](\{d_e\},\{\alpha_e\})dV \tag{6.36}$$

It is important to highlight that $[k_{IJ}]$ (5x5) is a submatrix of $[k_{Te}]$ (20x20), and in the same sense $[L_I]$ (4x5) is a submatrix used to assemble $[L_e]$ (4x20).

Once the matrices are integrated over the volume of the element, the system of equations for a finite element can be expressed as:

$$\begin{bmatrix} [k_{Te}] & [L_e]^T \\ [L_e] & [H_e] \end{bmatrix} \begin{bmatrix} \Delta\{d_e\} \\ \Delta\{\alpha_e\} \end{bmatrix} = \begin{bmatrix} \{f_e^{ext}\} - \{f_e^{int}\} \\ -\{h_e\} \end{bmatrix} = \begin{bmatrix} -\{r_e\} \\ -\{h_e\} \end{bmatrix} \tag{6.37}$$

Equation (6.31) includes the definition of the tangent stiffness matrix on element level $[k_{Te}]$ that can be split in three well defined matrices:

$$[k_{Te}] = [k_0] + [k_L(\{d\})] + [k_\sigma(\{\mathrm{S}\})] \tag{6.38}$$

where the stiffness matrix $[k_0]$ is known as the linear stiffness matrix, defined as in (6.45). The matrix addition of $[k_0]$ and $[k_L(\{d\})]$ is given as:

$$[k_0] + [k_L(\{d\})]_{IJ} = \int_{V_e} \left([B_I(\{d_I\})]^T [C] [B_J(\{d_I\})] \right) dV \tag{6.39}$$

and finally the stress stiffness matrix $[k_\sigma(\{\mathrm{S}\})]$:

$$[k_\sigma(\{\mathrm{S}\})] = [G(\{\mathrm{S}\})] = \int_{V_e} [G_{IJ}(\{S\})] dV \tag{6.40}$$

From (6.37) a static condensation of the internal degrees of freedom can be done. The second equation of the system leads to:

$$\Delta\{\alpha_e\} = \left(-\{h_e\} - [L_e]\Delta\{d_e\}\right) \{H_e\}^{-1} \tag{6.41}$$

which can be used to recover the incompatible degrees of freedom for the strain-stress post-processing.

If this result is used in the first equation of (6.37):

$$[k_{Te}]\Delta\{d_e\} + [L_e]^T [H_e]^{-1} \left(\Delta\{\alpha\} - [L_e]\Delta\{d_e\}\right) = \{f_e^{ext}\} - \{f_e^{int}\} = -\{r_e\} \tag{6.42}$$

Then, reordering terms:

$$\underbrace{\left([k_{Te}] - [L_e]^T [H_e]^{-1} [L_e]\right)}_{Condensated\ [k_{Te}]} \Delta\{u_e\} = \underbrace{\{f_e^{ext}\} - \{f_e^{int}\} - [L_e]^T [H_e]^{-1} \{h_e\}}_{Condensated\ \{r_e\}} \tag{6.43}$$

In order for all the stiffness matrices, load vectors, compatible displacements and incompatible displacements to be coherent they need to be defined in the same basis. In this formulation this basis is set to be the local orthonormal basis. As explained in chapter 4, in order to have quantities expressed in an orthonormal space, covariant components need to be multiplied by contravariant base vectors or vice versa.

This acquires more importance in section 6.5 in order to compute the stiffness matrices in the correct base.

6.4 Linear static analysis

As seen in the previous section, the geometrically non-linear analysis is quite complex to solve and computationally expensive. Therefore if some criteria are met, the problem can be solved linearly. Hence, easier and faster to solve.

In the case of being it correct to assume small strains, small rotations and small deformations on the definition of a problem, the quadratic term of the Green-Lagrange strain tensor (6.20) can be ignored. Thus, making the definition of strain linear.

Deeply explained in Cook et al. (2001, chap. 4) and Lund and Lindgaard (2016, pp. 42), by making use of a simpler variational approach accounting only for the linear interpolations of the strain displacement matrix $[B_I] = [B_0(\{G_i\})]$, any structure under loading or deformation reaches equilibrium when the potential energy of the system becomes stationary.

Summed up, the governing equation to be solved for linear static analysis is:

$$[K_0]\{D\} = \{F_{ext}\} \tag{6.44}$$

in which $\{F_{ext}\}$ is the global consistent vector of external loads and $[K_0]$ is the element linear stiffness matrix $[k_e]$ expanded to a global level. This last matrix is defined for the linear case as follows:

$$[k_e] = \int_{V_e} ([B_0(\{G_i\})]^T [C][B_0(\{G_i\})])dV \tag{6.45}$$

6.5 Numerical implementation

In this section, the calculation of the stiffness matrices and load vectors defined from equation (6.31) to (6.36) is done for the 4-node shell element.

6.5.1 Linear strain-displacement matrix

From (6.31) three different stiffness matrices need to be calculated which are introduced in (6.38). The first term of equation (6.31) represents the sum of the linear stiffness $[k_0]$ and the displacement stiffness matrix $[k_l(\{d\})]$. This expression depends on the current displacements and it is one of the reasons why an iterative process needs to be performed to solve the non-linear problem.

As explained in subsection 6.3.1, equation (6.20), the calculation of the non-linear strain displacement matrix $[B_I]$ can be done as the summation of a linear part that depends only on the original position of the nodes and a displacement related part that depends on the current displacements. In this subsection, the calculation of the linear strain-displacement matrix $[B_0(\{G_i\})]$, which is also used for the linear implementation of the 4-node shell element, is described.

Taking the linear part of the covariant strain expression (6.20) and evaluating the first variation

as in (5.12), the following is obtained:

$$\tilde{\varepsilon}_{ij} = \frac{1}{2}\left(\{G_i\}^T \cdot \frac{\partial \{u\}}{\partial x_j} + \{G_j\}^T \cdot \frac{\partial \{u\}}{\partial x_i} \right) \tag{6.46}$$

$$\delta\tilde{\varepsilon}_{ij} = \frac{1}{2}\left(\{G_i\}^T \cdot \frac{\partial \delta\{u\}}{\partial x_j} + \{G_j\}^T \cdot \frac{\partial \delta\{u\}}{\partial x_i} \right) \tag{6.47}$$

The numerical implementation presented in the following lines is intended to minimize the number of loops which is a desired feature for MATLAB. In the current version of the Fortran90 based MUST platform, this implementation is followed due to the previously coded linear element. It would increase the computational efficiency of the element to follow a more loop-wise approach.

The first step on the numerical implementation is to calculate the components needed in (6.47) and order them in the appropriate way. This is done with the matrix $[H]$:

$$\begin{Bmatrix} \tilde{\varepsilon}_1 \\ \tilde{\varepsilon}_2 \\ 0 \\ \tilde{\gamma}_{12} \\ \tilde{\gamma}_{23} \\ \tilde{\gamma}_{13} \end{Bmatrix}_I = [H] \begin{Bmatrix} \{G_1^T\} \cdot \frac{\partial \{u\}}{\partial r} \\ \{G_2^T\} \cdot \frac{\partial \{u\}}{\partial s} \\ \{G_3^T\} \cdot \frac{\partial \{u\}}{\partial t} \\ \{G_1^T\} \cdot \frac{\partial \{u\}}{\partial s} \\ \{G_2^T\} \cdot \frac{\partial \{u\}}{\partial r} \\ \{G_2^T\} \cdot \frac{\partial \{u\}}{\partial t} \\ \{G_3^T\} \cdot \frac{\partial \{u\}}{\partial s} \\ \{G_1^T\} \cdot \frac{\partial \{u\}}{\partial t} \\ \{G_3^T\} \cdot \frac{\partial \{u\}}{\partial r} \end{Bmatrix}_I \quad \text{where} \quad [H] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{6.48}$$

Now, in order to obtain the components that are multiplied by the matrix $[H]$ two more matrices are defined. The first one is a matrix containing the covariant vectors $\{G_i\} = \{G_i^x, G_i^y, G_i^z\}^T$ ordered in a suitable position:

$$\begin{Bmatrix} \tilde{\varepsilon}_1 \\ \tilde{\varepsilon}_2 \\ 0 \\ \tilde{\gamma}_{12} \\ \tilde{\gamma}_{23} \\ \tilde{\gamma}_{13} \end{Bmatrix}_I = [H][G] \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial t} \\ \frac{\partial w}{\partial r} \\ \frac{\partial w}{\partial s} \\ \frac{\partial w}{\partial t} \end{Bmatrix}_I \quad \text{where} \quad [G] = \begin{bmatrix} G_1^x & 0 & 0 & G_1^y & 0 & 0 & G_1^z & 0 & 0 \\ 0 & G_2^x & 0 & 0 & G_2^y & 0 & 0 & G_2^z & 0 \\ 0 & 0 & G_3^x & 0 & 0 & G_3^y & 0 & 0 & G_3^z \\ 0 & G_1^x & 0 & 0 & G_1^y & 0 & 0 & G_1^z & 0 \\ G_2^x & 0 & 0 & G_2^y & 0 & 0 & G_2^z & 0 & 0 \\ 0 & 0 & G_2^x & 0 & 0 & G_2^y & 0 & 0 & G_2^z \\ 0 & G_3^x & 0 & 0 & G_3^y & 0 & 0 & G_3^z & 0 \\ 0 & 0 & G_1^x & 0 & 0 & G_1^y & 0 & 0 & G_1^z \\ G_3^x & 0 & 0 & G_3^y & 0 & 0 & G_3^z & 0 & 0 \end{bmatrix} \tag{6.49}$$

Finally, the displacement derivatives can be calculated using:

$$\begin{Bmatrix} \tilde{\varepsilon}_1 \\ \tilde{\varepsilon}_2 \\ 0 \\ \tilde{\gamma}_{12} \\ \tilde{\gamma}_{23} \\ \tilde{\gamma}_{13} \end{Bmatrix}_I = [H][G][Q]_I \begin{Bmatrix} u_I \\ v_I \\ w_I \\ \alpha_I \\ \beta_I \end{Bmatrix}, \; [Q]_I = \begin{bmatrix} N_{I,r} & 0 & 0 & -\frac{t}{2}h_I N_{I,r}\{V_{21}^I\} & \frac{t}{2}h_I N_{I,r}\{V_{11}^I\} \\ N_{I,s} & 0 & 0 & -\frac{t}{2}h_I N_{I,s}\{V_{21}^I\} & \frac{t}{2}h_I N_{I,s}\{V_{11}^I\} \\ 0 & 0 & 0 & -\frac{1}{2}h_I N_I\{V_{21}^I\} & \frac{1}{2}h_I N_I\{V_{11}^I\} \\ 0 & N_{I,r} & 0 & -\frac{t}{2}h_I N_{I,r}\{V_{22}^I\} & \frac{t}{2}h_I N_{I,r}\{V_{12}^I\} \\ 0 & N_{I,s} & 0 & -\frac{t}{2}h_I N_{I,s}\{V_{22}^I\} & \frac{t}{2}h_I N_{I,s}\{V_{12}^I\} \\ 0 & 0 & 0 & -\frac{1}{2}h_I N_I\{V_{22}^I\} & \frac{1}{2}h_I N_I\{V_{12}^I\} \\ 0 & 0 & N_{I,r} & -\frac{t}{2}h_I N_{I,r}\{V_{23}^I\} & \frac{t}{2}h_I N_{I,r}\{V_{13}^I\} \\ 0 & 0 & N_{I,s} & -\frac{t}{2}h_I N_{I,s}\{V_{23}^I\} & \frac{t}{2}h_I N_{I,s}\{V_{13}^I\} \\ 0 & 0 & 0 & -\frac{1}{2}h_I N_I\{V_{23}^I\} & \frac{1}{2}h_I N_I\{V_{13}^I\} \end{bmatrix} \tag{6.50}$$

Expanding the matrix $[Q_I]$ as:

$$[Q]_{expanded} = [[Q_1] \; [Q_2] \; [Q_3] \; [Q_4]] \tag{6.51}$$

the $[B_0(\{G_i\})]$ matrix at the element level is obtained:

$$[B_0(\{G_i\})] = [H][G][Q]_{expanded} \tag{6.52}$$

6.5.2 Non-linear strain displacement matrix

The non-linear strain displacement matrix $[B_{NL}(\delta\{d\})]$ needs to calculate the contribution of the non-linear part of (6.20) $\eta_{ij} = \frac{1}{2}\frac{\partial\{u\}}{\partial x_i} \cdot \frac{\partial\{u\}}{\partial x_j}$. As with the linear part, the variation of the non-linear part is calculated:

$$\begin{Bmatrix} \delta\eta_1 \\ \delta\eta_2 \\ 0 \\ \delta\eta_{12} \\ \delta\eta_{23} \\ \delta\eta_{13} \end{Bmatrix} = \frac{1}{2}\left(\frac{\partial\{u\}}{\partial x_i} \cdot \frac{\partial\{\delta u\}}{\partial x_j} + \frac{\partial\{\delta u\}}{\partial x_i} \cdot \frac{\partial\{u\}}{\partial x_j}\right) \tag{6.53}$$

If the different terms of (6.53) are compared to the ones in (6.47) it is seen that the only difference is that the covariants are changed by displacement derivative terms. This means that, as suggested at the beginning of the chapter, the non-linear strain displacement matrix can be calculated as the linear one but with current displacements as input instead of positions of the nodes.

Thus, the only modification is presented in (6.49) where the terms of the displacement derivatives are computed instead of the derivatives of positions as done for the covariants. In this case, the derivative terms can be calculated by using the matrix $[Q_{expanded}]$ times the current

nodal displacements and using the resulting values in the, hereby defined $\{g\}_m$ vector:

$$\{g\}_m = \begin{Bmatrix} \{g_1\}_m \\ \{g_2\}_m \\ \{g_3\}_m \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial t} \\ \frac{\partial w}{\partial r} \\ \frac{\partial w}{\partial s} \\ \frac{\partial w}{\partial t} \end{Bmatrix} = [Q]_{expanded}\{d_{current}\} \tag{6.54}$$

Finally, the non-linear strain-displacement matrix is defined as:

$$[B_{NL}(\{d\})] = [H]\{g\}_m[Q_{expanded}] \tag{6.55}$$

6.5.3 Introduction of the Mixed Interpolated Tensorial Components

The implementation of the MITC enhancement explained in section 3.2 is done in the same way as the calculation of the linear and the non-linear strain displacement matrices and, the final result is substituted in the already calculated $[B_0(\{G_i\})]$ and $[B_{NL}(\{d\})]$.

Now, the transverse shear strains need to be calculated following the re-interpolation shown in (3.7).

$$\begin{Bmatrix} \tilde{\gamma}_{23} \\ \tilde{\gamma}_{13} \end{Bmatrix}_I = \begin{bmatrix} (1-s)(\{g_3\}^{B^T} N^B_{I,r} + \{g_1\}^{B^T} N^B_{I,t}) + (1+s)(\{g_3\}^{D^T} N^D_{I,r} + \{g_1\}^{D^T} N^D_{I,t}) \\ (1-r)(\{g_3\}^{A^T} N^A_{I,s} + \{g_2\}^{A^T} N^A_{I,t}) + (1+r)(\{g_3\}^{C^T} N^C_{I,s} + \{g_2\}^{C^T} N^C_{I,t}) \end{bmatrix} \{d_I\} \tag{6.56}$$

which can be computed matrix-wise by means of the following product of matrices:

$$\begin{Bmatrix} \tilde{\gamma}_{23} \\ \tilde{\gamma}_{13} \end{Bmatrix}_I = [H_{MITC}] \begin{Bmatrix} \{g_3\}^{B^T} N^B_{I,r} \\ \{g_1\}^{B^T} N^B_{I,t} \\ \{g_3\}^{D^T} N^D_{I,s} \\ \{g_1\}^{D^T} N^D_{I,t} \\ \{g_3\}^{A^T} N^A_{I,r} \\ \{g_2\}^{A^T} N^A_{I,t} \\ \{g_3\}^{C^T} N^C_{I,s} \\ \{g_2\}^{C^T} N^C_{I,t} \end{Bmatrix} \{d_I\} \tag{6.57}$$

where

$$[H_{MITC}] = \begin{bmatrix} 0 & 0 & (1-r) & (1-r) & 0 & 0 & (1+r) & (1+r) \\ (1-s) & (1-s) & 0 & 0 & (1+s) & 0 & (1+s) & 0 \end{bmatrix} \tag{6.58}$$

Similarly to previous calculations, the $[B_{MITC}]$ matrix containing the components to be introduced in the already defined $[B_0(\{G_i\})]$ and $[B_{NL}(\delta\{d\})]$ is calculated as:

$$[B_{MITC}] = [H_{MITC}][G_{MITC}][Q_{MITC}]_{expanded} \tag{6.59}$$

The calculation of the linear and non-linear parts of the enhancement is done in the same way as with $[B_0(\{G_i\})]$ and $[B_{NL}(\delta\{d\})]$ and the only change between them is the matrix $[G_{MITC}]$ containing the covariants.

All the matrices are described in appendix B.

6.5.4 Stress stiffness matrix

The matrix $[G_{IJ}]$ in expression (6.31) corresponds to the stress stiffness matrix. The calculation of this stiffness is determined by the expression

$$U_m = \int_{V_0} \{\hat{S}\} : \Delta\delta\{\tilde{\varepsilon}^c\} dV = \int_{V_e} \sum_{I=1}^{4}\sum_{J=1}^{4} \delta\{u_I\}^T [G_{IJ}] \Delta\{u_J\} dV \tag{6.60}$$

and the MITC enhancements. Expression (6.60) stands for the membrane strain energy as explained in Cook et al. (2001).

Following (5.12) and (5.18) while introducing the MITC interpolation terms from (3.7), the strain variations can be expressed as:

$$\begin{Bmatrix} \Delta\delta\tilde{\varepsilon}_{11} \\ \Delta\delta\tilde{\varepsilon}_{22} \\ \Delta\delta\tilde{\varepsilon}_{33} \\ \Delta\delta\tilde{\varepsilon}_{12} \\ \Delta\delta\tilde{\varepsilon}_{23} \\ \Delta\delta\tilde{\varepsilon}_{13} \end{Bmatrix}^c = \begin{Bmatrix} \delta\{g_1\} \cdot \Delta\{g_1\} \\ \delta\{g_2\} \cdot \Delta\{g_2\} \\ 0 \\ \delta\{g_1\} \cdot \{g_2\} + \delta\{g_2\} \cdot \{g_1\} \\ \frac{1}{2}(1+r)\Delta\delta\tilde{\varepsilon}^D_{st} + \frac{1}{2}(1-r)\Delta\delta\tilde{\varepsilon}^B_{st} \\ \frac{1}{2}(1+s)\Delta\delta\tilde{\varepsilon}^A_{rt} + \frac{1}{2}(1-s)\Delta\delta\tilde{\varepsilon}^C_{rt} \end{Bmatrix} \tag{6.61}$$

The goal now is to find a way to formulate a $[G_{IJ}]$ that when multiplied by the displacements of the nodes I and J displays the membrane strain energy U_m associated with (6.60). The problem lies in not being able to use the general formulation presented in sources like Cook et al. (2001) due to the MITC enhancements. However, the solid shell formulation presented in Klinkel et al. (1999) can be used as inspiration but needs to be modified because of the addition of the rotational degrees of freedom.

One important feature of the stiffness matrix $[G_{IJ}]$ is the fact that each sub-matrix is multiplied by the displacement terms d_I and d_J. This property also holds for the components of each sub-matrix. This is used to build the formulas to calculate each term of the stiffness matrix according to what displacement will they be multiplied by.

Another consideration is the way the shell element is formulated regarding the degrees of freedom. For each node there are 5 degrees of freedom (disregarding the 6^{th} one introduced afterwards with a penalty method). The first three are translational and the last two rotational. This has a big influence on the calculation of the covariants in the expression (6.61) as the

derivatives are not simply the derivatives of the isoparametric shape functions. To take this into account, the stiffness matrix is separated in blocks as follows:

$$\left[\begin{array}{c|c|c|c} \hat{G}_{11} & \hat{G}_{12} & \hat{G}_{13} & \hat{G}_{14} \\ \hline \hat{G}_{12} & \hat{G}_{22} & \hat{G}_{23} & \hat{G}_{24} \\ \hline \hat{G}_{13} & \hat{G}_{23} & \hat{G}_{33} & \hat{G}_{34} \\ \hline \hat{G}_{14} & \hat{G}_{24} & \hat{G}_{34} & \hat{G}_{44} \end{array}\right] \left\{\begin{array}{c} \{d_1\} \\ \{d_2\} \\ \{d_3\} \\ \{d_4\} \end{array}\right\} \quad \text{where} \quad \{d_I\} = \left\{\begin{array}{c} u_I \\ v_I \\ w_I \\ \alpha_I \\ \beta_I \end{array}\right\} \tag{6.62}$$

Thus, allowing to properly multiply each group of nodal degrees of freedom by the appropriate stiffness values. The matrices $[\hat{G}_{IJ}]$ are formed in the following way:

$$[\hat{G}_{IJ}] = \begin{bmatrix} \hat{S}_{IJ} & 0 & 0 & 0 & 0 \\ 0 & \hat{S}_{IJ} & 0 & 0 & 0 \\ 0 & 0 & \hat{S}_{IJ} & 0 & 0 \\ 0 & 0 & 0 & \hat{A}_{IJ} & 0 \\ 0 & 0 & 0 & 0 & \hat{B}_{IJ} \end{bmatrix} \tag{6.63}$$

Each of these constants incorporates the contributions from calculating the scalar products needed for the derivatives of (6.61). The three $\hat{S}_{IJ}$ correspond to the isoparametric translations, while the $\hat{A}_{IJ}$ and $\hat{B}_{IJ}$ corresponds to the rotational degrees of freedom. In order to understand the following calculation it is useful to remember that what is being done is obtaining the derivatives of the displacement field. This derivatives have contributions from the original isoparametric formulation and the enhancements. After having these contributions, their placement in the stiffness matrix allow them to be combined creating the strain terms. From this idea, the following terms are defined:

$$\begin{aligned} \hat{S}_{IJ} &= \hat{S}^m_{IJ} + \hat{S}^s_{IJ} \\ \hat{A}_{IJ} &= \hat{A}^m_{IJ} + \hat{A}^s_{IJ} \\ \hat{B}_{IJ} &= \hat{B}^m_{IJ} + \hat{B}^s_{IJ} \end{aligned} \tag{6.64}$$

Appendix C provides the formulas to do all the calculations of the stress stiffness matrix.

6.5.5 Numerical integration

The purpose of numerical integration, also called Gauss quadrature, is to numerically assemble the stiffness matrix. There are many quadrature rules, each conducts a sampling of the function at specific points (Gauss points), multiplying the resulting number by an appropriate weight factor and adding up the results (Cook et al., 2001, chap. 6).

One layer integration

The element used in this work uses a three dimensional Gauss rule combining a 2x2 quadrature requiring four in-plane sampling points and two Gauss points in the layer thickness direction.

An example of how it is implemented in this work can be shown by using the definition of the linear stiffness matrix (6.45) of a one layer finite element:

$$\begin{aligned}[k_e] &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}[B_0(\{G_i\})]^T[C][B_0(\{G_i\})]\ |J|\ h\ dr\ ds\ dt \\ &= \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} W_i\ W_j\ W_k[B_0(\{G_i\})]^T[C][B_0(\{G_i\})]\ |J|\ h \end{aligned} \tag{6.65}$$

where W_i, W_j and W_k are the weighting factors (equal to one for the 2x2 Gauss rule), $|J|$ is the determinant of the Jacobian of the interpolation functions and h is the layer thickness.

The fact that the numerical integration is of a 2x2 quadrature means that the element formulated in this thesis uses a full integration rule. The coordinates of the Gauss points according to such a rule are seen in figure 6.7, and the value of their respective weighting factor is equal to one.

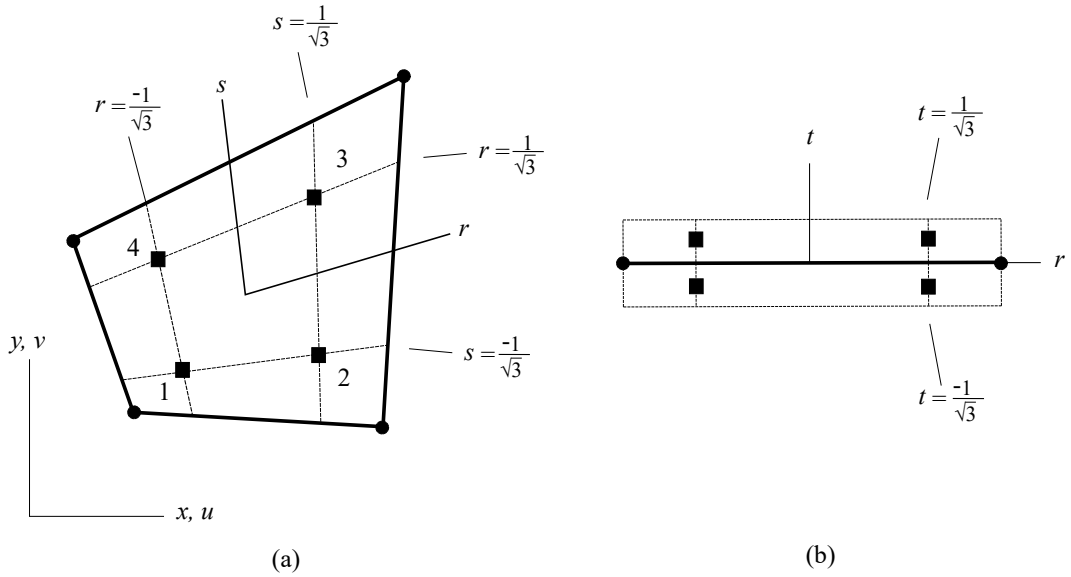


Figure 6.7. Sampling point locations for integration of the shell element interpolation functions using Gauss rule of order 2 both in-plane (a) and through-thickness (b).

Multi-layer through-the-thickness integration

As the element needs to be prepared to model laminated structures, the interaction between laminae is found by means of thickness integration. There are two different methods to integrate over the laminate thickness: the explicit integration (Lund, 2016a, pp. 250) and the layer-wise numerical integration. This section focuses on the layer-wise numerical integration because it is the one used to code the 4-node shell element of this work.

The methodology consists on regarding each layer as a sub-element of the laminated element. Therefore, we integrate over each layer, and for each layer, we integrate the whole volume of the sub-element as explained in the previous subsection. In order to do so, a change of variable needs to be done so that the integration limits correspond to $[\pm 1]$ and the Gauss quadrature can be implemented.

The change is to substitute the element thickness variable t with the thickness coordinate t_l:

$$t = -1 + \frac{1}{h}\left(2\sum_{i=1}^{l} h_i - h_l(1 - t_l)\right) \tag{6.66}$$

where h_l is the thickness of the l'th layer.

With the variable change in (6.66), the differential of thickness dt can be obtained, leading to:

$$dt = \frac{h_l}{h} dt_l \tag{6.67}$$

Now, if this change of variable is applied to equation (6.65), the stiffness matrix for each layer of the element is:

$$[k_e]_{layer} = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} W_i\ W_j\ W_k [B_0(\{G_i\})]^T [C][B_0(\{G_i\})]\ |J|\ \frac{h_l}{h} \tag{6.68}$$

Although it is the method used in this work it is computationally expensive for layered elements with many layers.

6.5.6 Strain and stress post-processing

This subsection presents the strain and stress post-processing once the numerical results for the displacements are obtained. It is dealt in the following, where the strains and stresses are evaluated in order to get accurate results and their extrapolation to the nodes. This subsection also presents the outline of this post-processing whether the analysis is linear or geometrically non-linear.

Strain and stress at the superconvergent points and node extrapolation

Strain fields are likely to display greater error than the displacement field. This is because they are obtained by differentiation of an approximated displacement field. Along with that, the strains and stresses are usually more accurate at a location within the element rather than the element boundaries. It is shown in some elements that Gauss points of lower order rules than the used to compute the stiffness matrix display better results for strains (Cook et al., 2001, chap. 6). These are called superconvergent points.

The element of this work evaluates the strains at the superconvergent points, which are a 2x2 quadrature (defined in figure 6.7) for the in-plane components (enhanced with the EAS, which increases the order of the strain interpolation), and one superconvergent point in the element centre for the transverse shear components enhanced with the MITC method.

From their sampling point, the strain and stress components are extrapolated to the nodes. The transverse shear components γ_{yz} and γ_{xz} are evaluated at one point in the element centre, therefore considered constant all along the element. As explained in chapter 2, the transverse normal strain ε_{zz} is assumed to be zero. In the case of the in-plane components ε_{xx}, ε_{yy} and γ_{xy} their values are extrapolated by setting a new set of coordinates at each Gauss point.

In order to extrapolate the 4 Gauss point values to the 4 nodes at the plane of the corresponding thickness Gauss point, dimensionless coordinates (r_{loc}, s_{loc}) that are proportional to (r, s) are introduced. Figure 6.8 graphically describes this coordinate system.

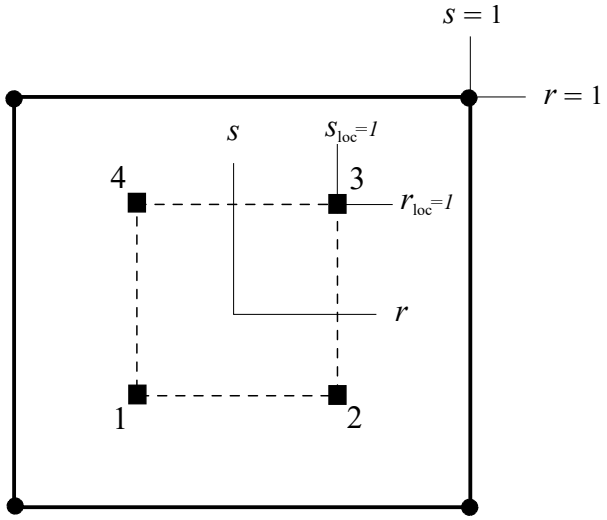


Figure 6.8. Reference coordinate systems (r_{loc}, s_{loc}) and (r, s) used in extrapolation of stresses from Gauss points. Squares show Gauss points of a 2x2 rule.

The absolute relation between coordinates systems is:

$$r_{loc} = \sqrt{3}r \quad \text{and} \quad s_{loc} = \sqrt{3}s \tag{6.69}$$

The strain or stress at any arbitrary point can be obtained by the usual shape functions formulated in the new coordinate system taking into account the relation in equation (6.69). In the in-plane case of the 4-node shell element developed in this work, the strain or stress value at any point is expressed as:

$$\{\varepsilon\}_I = \sum_{gp=1}^{4} N_{gp}\{\varepsilon_{gp}\} \tag{6.70}$$

where N_{gp} is the bilinear shape function of each Gauss point but replacing (r, s) by (r_{loc}, s_{loc}) and $\{\varepsilon_{gp}\}$ is the strain vector obtained in by following $\{\varepsilon\} = \{\varepsilon^c\} + \{\hat{\varepsilon}\}$ with the corresponding strain-displacement matrices evaluated at the superconvergent points.

Once the strain values are extrapolated to the corner nodes in the plane of the top and bottom thickness Gauss points, they are extrapolated to the top and bottom surface of the element by means of the same procedure and the same relation $t_{loc} = \sqrt{3}t$. This time, though, using the linear shape functions $N_1 = \frac{1}{2}(1 + t)$, $N_2 = \frac{1}{2}(1 - t)$ that interpolate the strain field through-the-thickness.

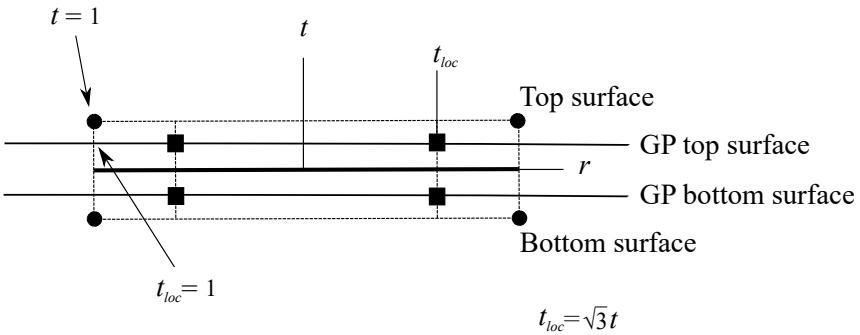


Figure 6.9. Reference coordinate systems (t_{loc}) and (t) used in extrapolation of stresses from Gauss points. Squares show Gauss points of a 2x1 rule.

Outline of the strain-stress post-processing algorithm

The outline of the post-processing is summarized in the flow chart corresponding to figure 6.10. This chart describes step by step the algorithm used to achieve the strain or stress values in the element level.

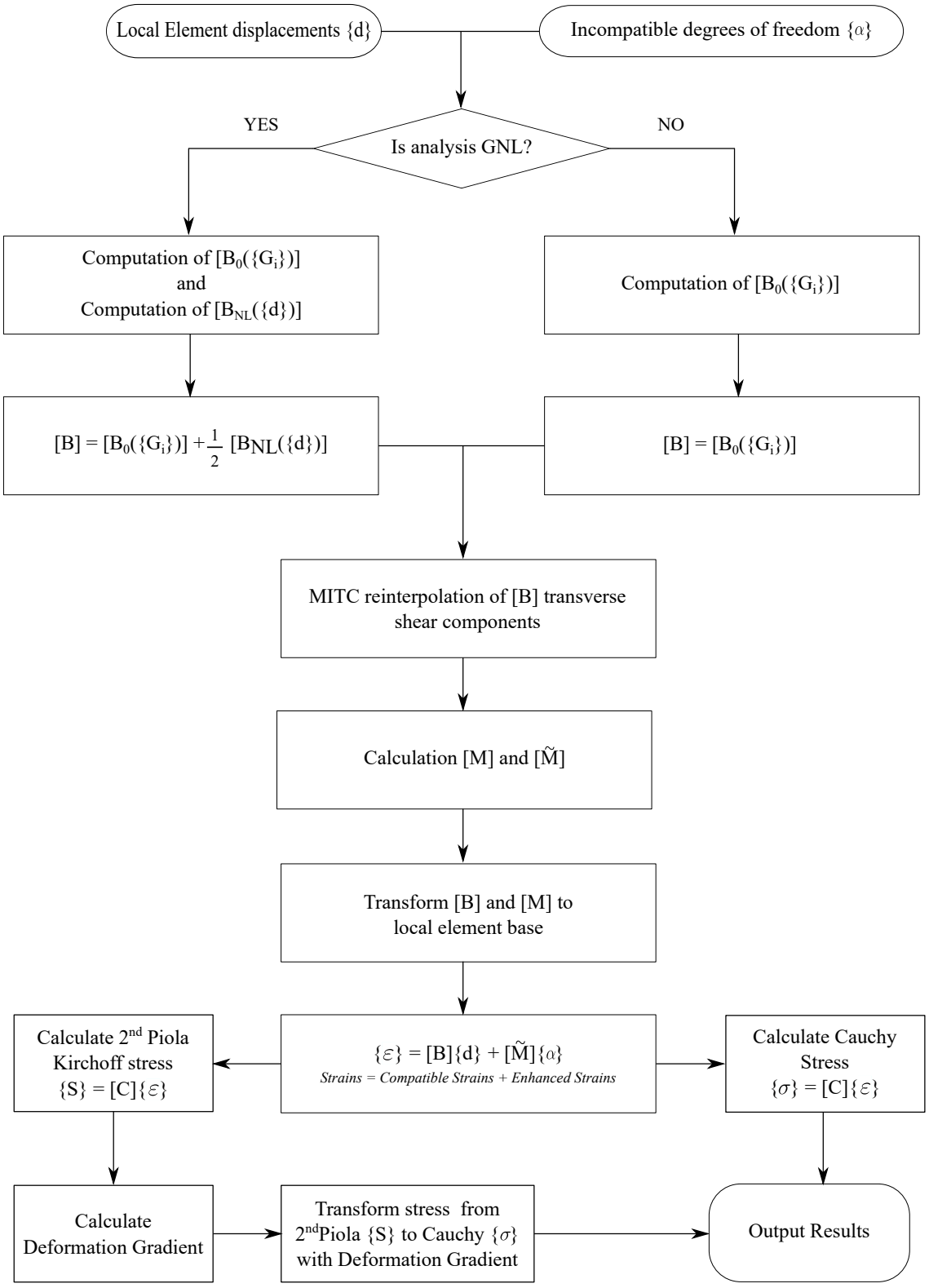


Figure 6.10. Flow chart of the algorithm used to post-process the displacements in order to get strains and stresses in element level.

6.5.7 Drilling degree of freedom

As mentioned in section 6.1 the degeneration process to formulate the kinematics of the 4-node shell element has only 5 degrees of freedom. The so called drilling degree of freedom needs to be introduced as it is of relevance when transforming the stiffness matrix from the local coordinates to the global ones. Moreover, the 6^{th} degree of freedom is needed when the elements are linked to other type of elements by means of MPC's or other methods.

In order to add the drilling degree of freedom the approach followed in Ravn et al. (2008) is applied. A penalty stiffness is added to the stiffness matrix. A block of the stiffness matrix is defined as being the one multiplying the degrees of freedom of one node:

$$\{f_{ext}\} = \left[\begin{array}{c|c|c|c} BLOCK & BLOCK & BLOCK & BLOCK \\ \hline BLOCK & BLOCK & BLOCK & BLOCK \\ \hline BLOCK & BLOCK & BLOCK & BLOCK \\ \hline BLOCK & BLOCK & BLOCK & BLOCK \end{array}\right] \begin{Bmatrix} \{d\}_1 \\ \{d\}_2 \\ \{d\}_3 \\ \{d\}_4 \end{Bmatrix} \tag{6.71}$$

where each block is defined as follows:

$$BLOCK = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{bmatrix} \tag{6.72}$$

To add the drilling degree of freedom a 6^{th} row and column need to be added so now a 6×6 matrix can be multiplied by a vector of 6 degrees of freedom per node. To do so, a penalty stiffness is added as proposed in (Ravn et al., 2008). The result can be seen in (6.73). This penalty stiffness has to be small enough in order to not change the results but big enough to avoid a singular stiffness matrix.

$$BLOCK = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & 0 \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & 0 \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & 0 \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{k_{44}+k_{55}}{1000} \end{bmatrix} \tag{6.73}$$

The new block matrices are then ensembled again to form a new 6 DOF stiffness matrix.

6.6 Linear Buckling

Buckling occurs when slender structures enter an unstable state after compressive loads are applied. Composite structures, such as wind turbine blades, are usually build of shells which

can be easily buckled and therefore, it is of great interest to quantify the maximum load that can be applied to the structure before it buckles. In this section an introduction of the buckling phenomenon is given in order to formulate the linear buckling problem in terms of the finite elements method.

The buckling phenomena can be understood from an energetic or a mathematical point of view. First the energetic point of view is introduced. In figure 6.11 a slender column is loaded with a compressive axial load. In the first stages of the loading process the column remains in a stable equilibrium situation and consequently, it is able to carry the load. In this process, elastic membrane energy is accumulated in the column, but when the load reaches a critical value known as buckling load, the membrane energy is suddenly transformed into bending energy without any extra work being applied. This induces a sudden change on the deformation state that could lead to a failure as the structure may not be able to carry the load any more. The buckling analysis only provides the value of critical load that produces buckling. If more information of the post-buckling deformation is required to assess whether the structure fails, a post-buckling analysis is required.

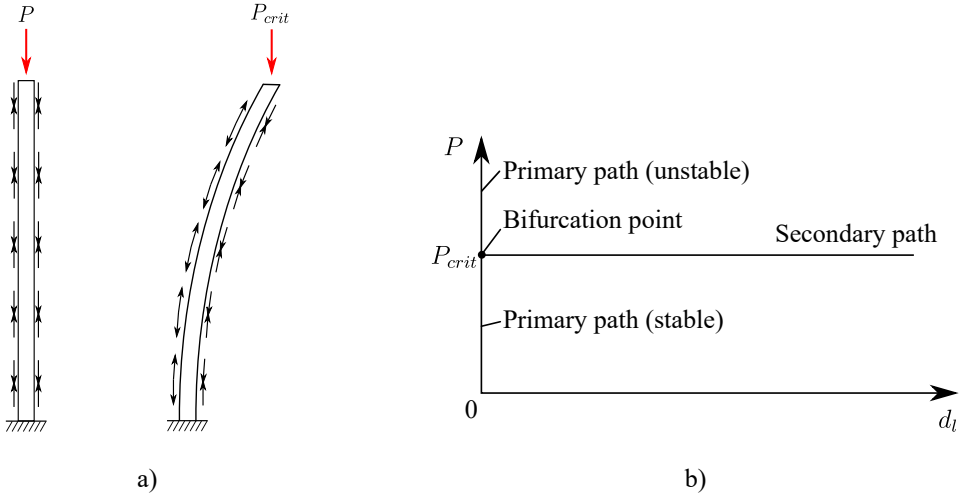


Figure 6.11. a) Representation of the membrane to bending energy transformation that occurs during buckling. b) Load (P) - Lateral displacement (d_l) theoretical path without material or loading imperfections.

Mathematically the buckling concept can be seen in the displacement-load curves in figure 6.11. The primary path corresponds to the column being loaded before buckling occurs. A stable equilibrium means that the structure is able to hold this position and carry the load as small imperfections or loading misalignments will have little, if any, effect on the deformation state. When the theoretical buckling load is achieved a singularity called bifurcation point is achieved. This singularity can be seen in the derivation of the theoretical Euler buckling Load. In Cook et al. (2001, chap. 18, sec. 1) a simple straight-forward derivation of the Euler buckling load by means of variational methods is given. At the end of the derivation, the expression of the lateral displacement is given as:

$$d_l = \frac{q_c L}{2([k_0] + [k_\sigma(\{S\})])} \tag{6.74}$$

where $[k_0]$ is the lateral bending associated stiffness and $[k_\sigma(\{S\})]$ is the membrane stress associated stiffness. When the forces are compressive $[k_\sigma(\{S\})]$ is actually making the effective lateral stiffness ($[k_0]+[k_\sigma(\{S\})]$) smaller. The mathematical singularity is then achieved when the

stress stiffness makes the effective lateral stiffness 0. The load value that causes this singularity is the so called theoretical buckling load, and bigger loads than this are mathematically possible but physically unachievable because, being an unstable equilibrium, any imperfection or misalignment of the load will make the structure collapse. In Dym and Shames (2013) a more detailed derivation of the Euler load can be found in case of interest.

The fundamental idea behind buckling load calculation, both analytically or by means of FEM and other numerical methods, is to calculate the strain energy associated to the lateral bending and the strain energy associated to the membrane stresses. After that, the value of compressive load that cancels out the bending associated energy has to be found.

Although it can seem that the calculation of the buckling load is straight forward, the theoretical buckling load only gives an upper bound to the magnitude the load P can achieve before buckling. The reason for that is found in the fact that one of the assumptions of the calculation of this theoretical load is that the load is perfectly aligned with the axis and the column or beam studied has no defects.

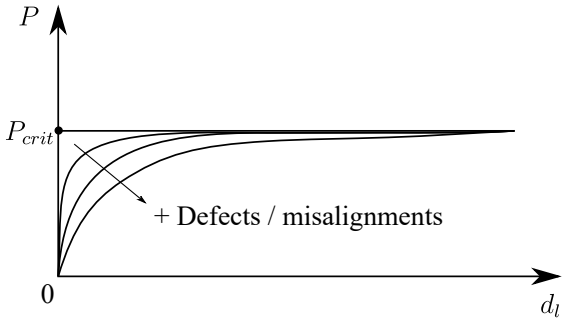


Figure 6.12. Load-displacement paths when the defects on the material or the load misalignment increase.

As can be seen in figure 6.12 the reality of the buckling phenomena is more complex than what has been explained and needs further analysis in order to avoid over estimating the buckling load and also to know what is exactly happening in the structure.

One of the main drawbacks of the buckling analysis done by means of a geometrically non-linear formulation, specially when optimizing a structure, is the computational power required. A solution for that is a linearization of the non-linear analysis for the buckling case reducing, though, the precision of the analysis and the relevant information that can be gathered from it.

The linear buckling analysis starts by applying a reference load $\{P_{ref}\}$ to the structure that is being analyzed. If the displacements are assumed to be small, linear behavior can be assumed for the stress stiffness matrix:

$$\{P\} = \lambda\{P_{ref}\} \implies [K_\sigma(\{S\})] = \lambda[K_\sigma(\{S\})]_{ref} \tag{6.75}$$

In addition the contribution of the initial displacement matrix $[K_L(\{D\})]$ can be disregarded leaving the expression of the equilibrium equation when a critical force is applied as:

$$([K_0] + \lambda_{crit}[K_\sigma(\{S\})]_{ref})\{D\}_{ref} = \lambda_{crit}\{P_{ref}\} \tag{6.76}$$

It is known that when buckling occurs there is a variation on the displacement state without any application of an additional external force:

$$([K_0] + \lambda_{crit}[K_\sigma(\{S\})]_{ref})\{D_{ref} + \delta D\} = \lambda_{crit}\{P_{ref}\} \tag{6.77}$$

Subtracting expression (6.76) from (6.77):

$$([K_0] + \lambda_{crit}[K_\sigma(\{S\})]_{ref})\{\delta D\} = \{0\} \tag{6.78}$$

By solving this eigenvalue problem, a set of eigenvalues λ_{crit} are obtained that multiplied by the reference load give the linear buckling load. Associated with each eigenvalue, the eigenvectors provide the buckling modes.

Thus, the linear buckling analysis is performed in the following way:

1. Perform a linear static analysis
2. Compute strains and stresses
3. Compute Stress Stiffness
4. Solve the eigenvalue problem

The MUST platform has an implemented solver for the eigenvalue problem of the linear buckling analysis: the subspace iteration algorithm. It is an iterative method for computing eigenspaces of large matrices. For more details on this algorithm check Saad (1992). In the case of solving the problem in Matlab, the internal solver called with the function $eig([K_0],-[K_\sigma(\{S\})]_{ref},''default\ algorithm'')$ is used.

Numerical Tests 7

In previous chapters the non-linear formulation for the 4-node shell element with the EAS and MITC enhancements has been derived. Once the MUST platform is running its implementation, numerical tests are carried out to assess the performance of the element. This chapter is divided in three distinguished sections in which different aspects of the work are checked. Firstly, the linear formulation obtained from the linearization of the governing equations in section 6 is checked for convergence and accuracy. Secondly, the geometrically non-linear element obtained in this work is assessed by studying different bench mark examples including various geometries and load cases. Finally, the chapter concludes by checking the behavior of the element under buckling loading conditions.

7.1 Numerical tests for the linear formulation

This section is based on the numerical tests in MacNeal and Harder (1985), which include membrane and bending patch tests as well as displacement precision tests under different loads and meshes. Thus, the linear 4-node enhanced element is evaluated on its reliability and performance. The use of patch tests provides also a source of determining whether the element is implemented correctly. As follows, the numerical tests introduced above are performed to:

- Linear isotropic elements
- Linear laminated elements

7.1.1 Tests for the linear isotropic element

Patch test

The patch test is used in this work as a standard methodology to check if the fully enhanced element converges. It was designed by Bruce Irons to test if a patch of elements was capable of modeling a proper constitutive behavior when subjected to constant strain or stress. As explained in Taylor et al. (1986), if the patch of elements exhibits the true behavior of the material, when the elements get infinitesimally small the model represents the real material and can converge to the exact solution.

In finite element anlysis, convergence is ensured if the following two criteria are fulfilled:

1. The displacement interpolations are able to model rigid body motion without introducing any strains.
2. The displacement interpolations are able to model a situation of constant strain or stress.

Being the last requirement fulfilled makes the first criterion effective automatically. Therefore, if the patch test is successfully passed, convergence of the element is guaranteed.

In order to properly test the enhanced 3D shell element of this project, two patch test are carried out. Both are composed of the same distorted mesh but different load cases so membrane or bending strains can be tested.

Mesh

As defined in MacNeal and Harder (1985) the mesh of the patch tests with the global nodal and element numbering can be seen in figure 7.1. Table 7.1 shows the position of the nodes in the global coordinate system.

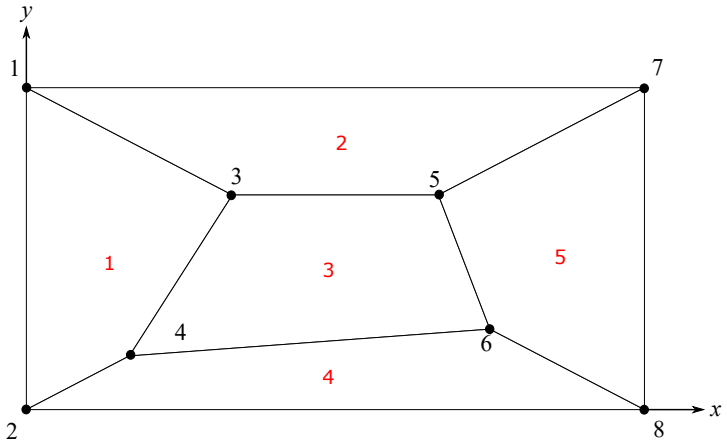


Figure 7.1. Patch test mesh with the global element numbering in red.

Node	x	y	*Node*	x	y
1	0	0.12	5	0.16	0.08
2	0	0	6	0.18	0.03
3	0.08	0.08	7	0.24	0.12
4	0.04	0.02	8	0.24	0

Table 7.1. Position of nodes.

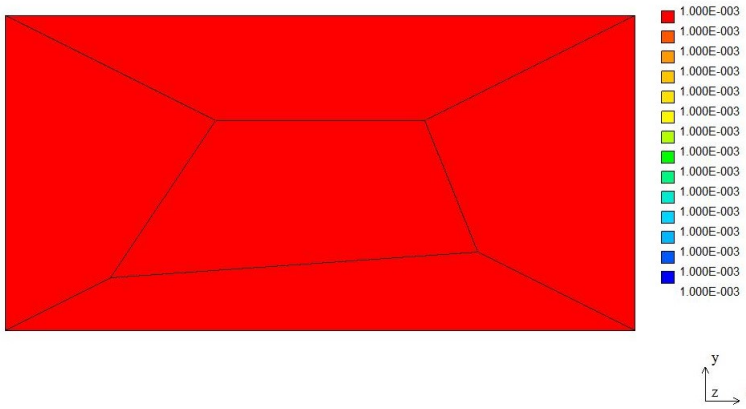
Membrane patch test for linear isotropic elements

According to MacNeal and Harder (1985) the following boundary conditions are applied to nodes 1, 2, 7 and 8 in order to obtain a constant normal and shear in-plane strain distribution:

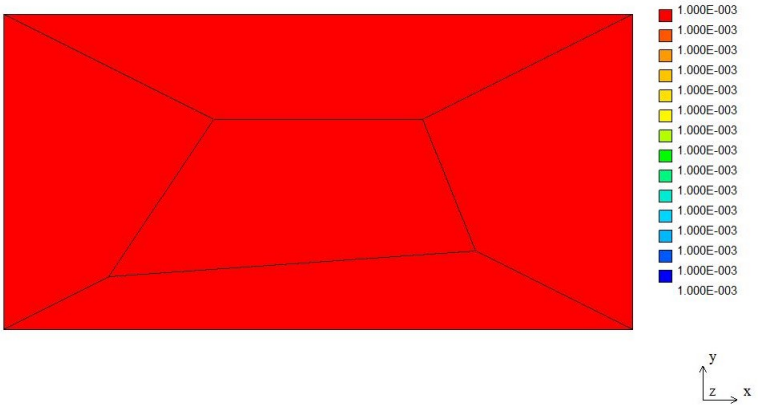
$$u = 10^{-3}(x + y/2) \qquad v = 10^{-3}(y + x/2) \tag{7.1}$$

which give a constant strain distribution of $\varepsilon_x = \varepsilon_y = \gamma_{xy} = 10^{-3}$. The material properties used are $E = 10[MPa]$ and $\nu = 0.3$.

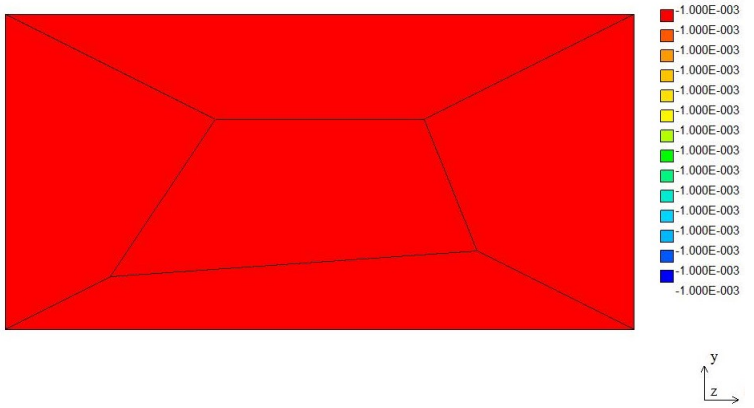
As seen in figure 7.2, the results show that the element passes the membrane patch test.



(a) ε_x strain field



(b) ε_y strain field



(c) γ_{xy} strain field

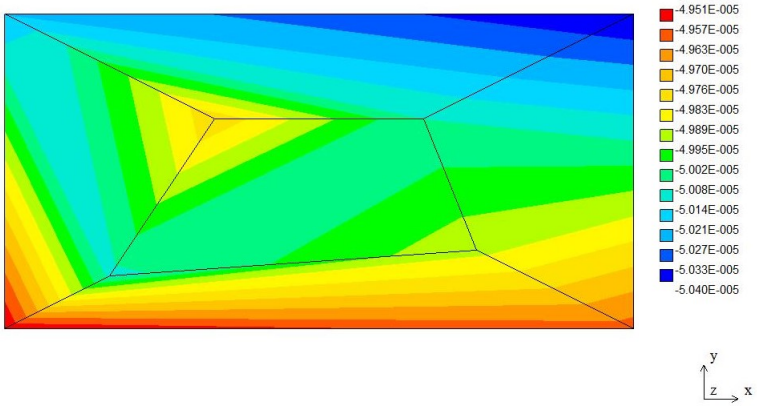
Figure 7.2. Membrane patch test results

Bending patch test for linear isotropic elements

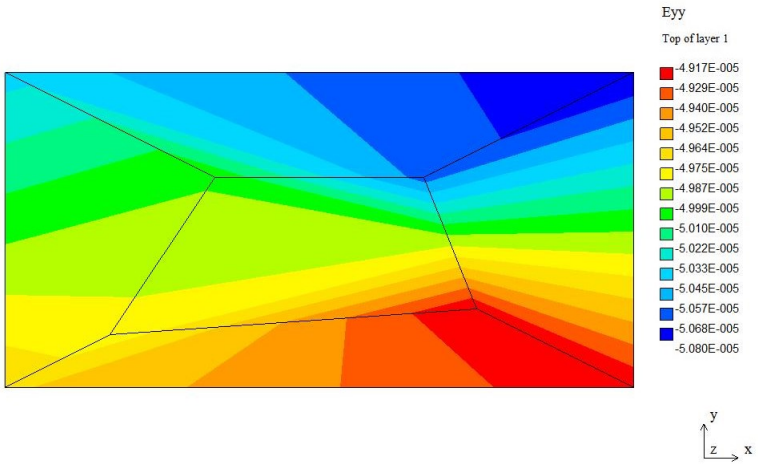
The same procedure as in the previous subsection is applied with boundary conditions in nodes 1, 2, 7 and 8 that ensure constant strains and bending moments:

$$w = 10^{-3}\left(\frac{x^2+xy+y^2}{2}\right) \qquad \theta_x = 10^{-3}\left(y+\frac{x}{2}\right) \qquad \theta_y = -10^{-3}\left(x+\frac{y}{2}\right) \tag{7.2}$$

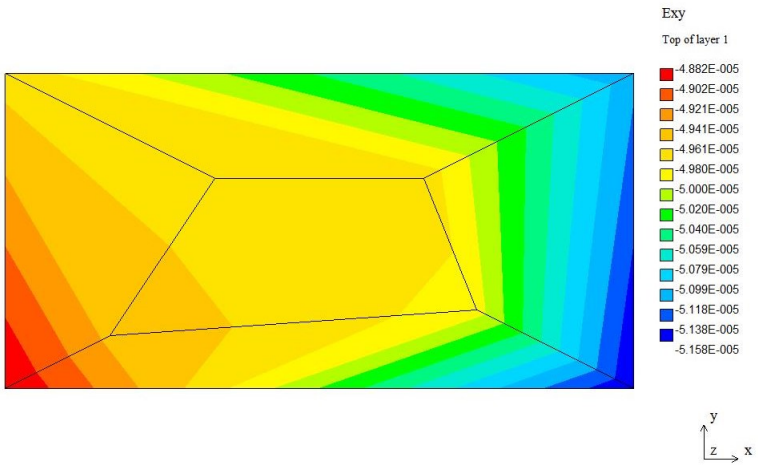
With the boundary conditions in (7.2) the results are shown in figure 7.3.



(a) ε_x strain field



(b) ε_y strain field



(c) γ_{xy} strain field

Figure 7.3. Bending patch test results

The patch test is passed successfully as $\varepsilon_x = \varepsilon_y = \gamma_{xy} = -5 \times 10^-5$ is expected from the theoretical calculations (MacNeal and Harder, 1985). The exact same inaccuracies appear when performing the patch test to the ANSYS element shell 181.

Bench marks

In this subsection various comparisons are done between different load conditions and meshes to assess the performance of the 4-node linear shell element enhanced by means of MITC and EAS. The comparison is done between the Fortran 90 coded element implemented for the MUST platform and the theoretical results. The problems are set up following MacNeal and Harder (1985) and consist of a cantilever beam with a mesh of 6 elements length-wise and one in height (figure 7.4).

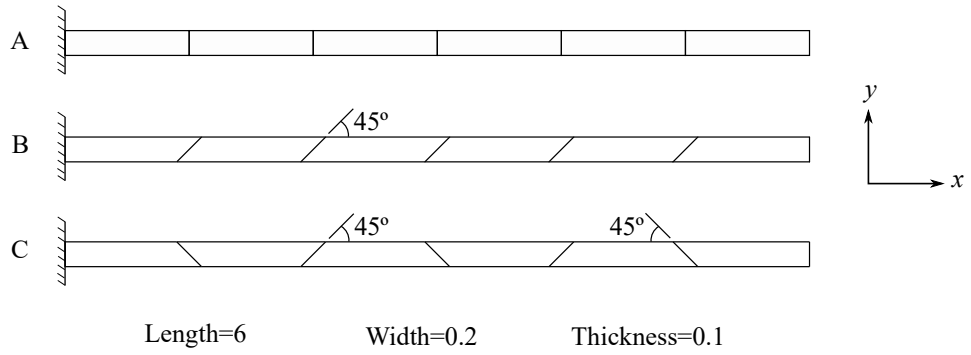


Figure 7.4. Element shapes investigated and dimensions of the cantilever beam.

The cantilever beam is loaded on the right free tip with 4 different load cases:

- Extensional load in the x direction.
- Load in the y direction (in-plane bending).
- Load in the z direction (transverse bending).
- Twisting load (x rotation direction).

All the loads are applied as unit forces at the tip nodes. The goal of this load cases is to test how the enhancements work in the element. The different meshes give information on the behaviour of the element when the mesh is not regular. This information is important in the case the element needs to mesh complicated geometry.

The theoretical results are given by MacNeal and Harder (1985), although in the twist case there is a typographic mistake that is corrected in this report.

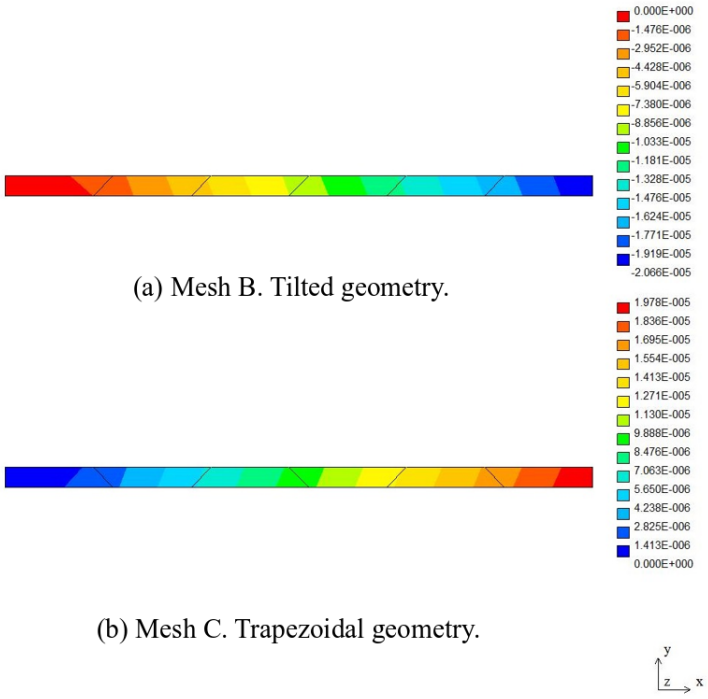
$$\begin{aligned} u_{extension} &= 3.0 \times 10^{-5}[m] & v_{in\text{-}plane\ bending} &= 0.1081[m] \\ w_{out\text{-}of\text{-}plane\ bending} &= 0.4321[m] & w_{twist} &= 320.91 \times 10^{-6}[m] \end{aligned} \tag{7.3}$$

All the results in (7.3) are maximum displacements (disregarding the sign) of one of the two nodes at the tip of the beam. The results are normalized with respect to the theoretical result from (7.3).

Table 7.2. Load cases normalized displacement wrt. the theoretical result.

Load case	Mesh	Linear Shell 4 MITC & EAS	ANSYS shell 181
Extension	A	0.996	0.997
	B	1.012*	1.223
	C	1.011*	1.207
Twist	A	0.940	0.941
	B	0.794 & 0.903	0.848 & 0.904
	C	1.078 & 0.689	0.689 & 1.078
In-plane shear	A	0.993	0.982
	B	0.632	0.626
	C	0.052	0.052
Out-of-plane shear	A	0.980	0.980
	B	0.978	0.979
	C	0.963	0.963

Table 7.2 shows good results for all load cases as long as the mesh is kept regular. When the mesh is regular, all the properties of the element are known and well defined, therefore, the element if correctly implemented, gives adequate results. Once the mesh is distorted some unexpected behavior is appreciated in the results. Marked with (*) are the apparently good results for the extension load case. Although they look accurate they display a displacement in the y direction of the same magnitude as the obtained displacement in the x direction as it can be seen in figure 7.5, which happens as well using the ANSYS 181 shell element.

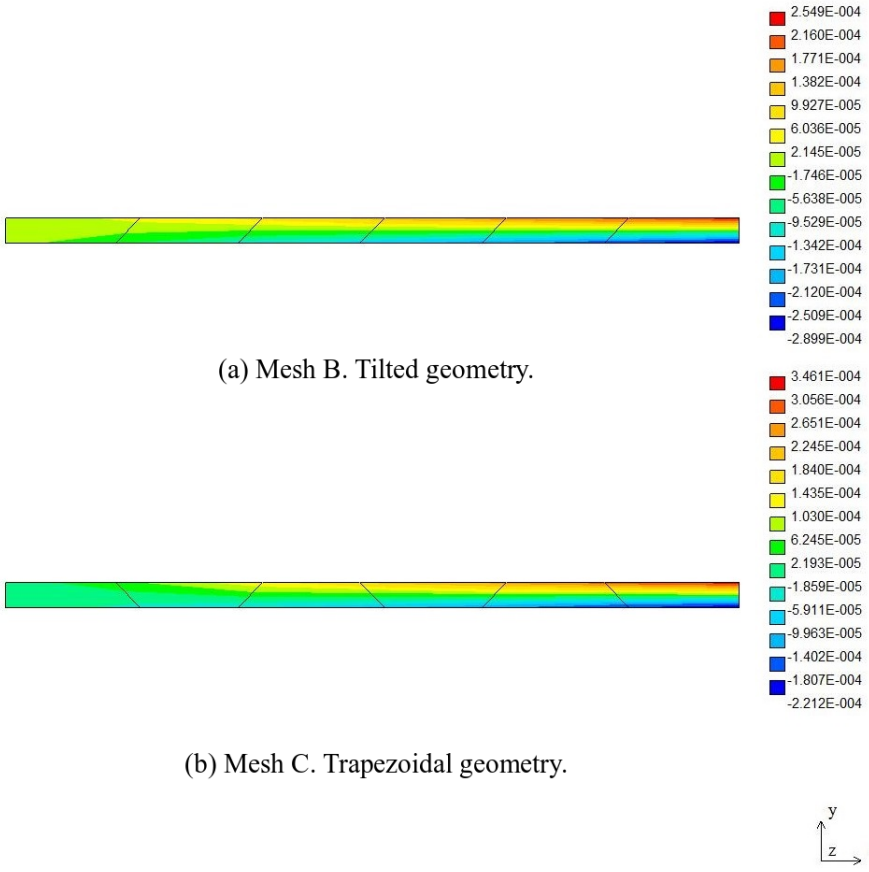


(a) Mesh B. Tilted geometry.

(b) Mesh C. Trapezoidal geometry.

Figure 7.5. Displacement v for the beam under a unit extensional load in the x direction.

Another set of results that present anomalies is the twisting load case. Once the mesh is distorted, the expected symmetry of the results vanishes leading to asymmetric results as seen in figure 7.6. Besides the asymmetry, the element also becomes over stiff (mesh B) or over compliant (mesh C) due to the geometry distortion.



(a) Mesh B. Tilted geometry.

(b) Mesh C. Trapezoidal geometry.

Figure 7.6. Displacement w for the beam under a unit twisting load in the tip.

7.1.2 Tests for the laminated linear element

Patch test

When testing the laminated linear element, the major test is to determine whether the piece-wise through-the-thickness integration is performed correctly. By carrying out the same patch tests as described earlier but with a section layup as in figure 7.7, the element is evaluated for its ability to model constant strain.

The total patch thickness is of 1 mm and it is modeled as three isotropic layers instead of one single equivalent layer. The thickness of each layer is 0.2 mm, 0.2 mm and 0.6 mm respectively. All the tests yield the exact same results as for the single layer case. Therefore it is shown that the piece-wise through-the-thickness integration is performed correctly.

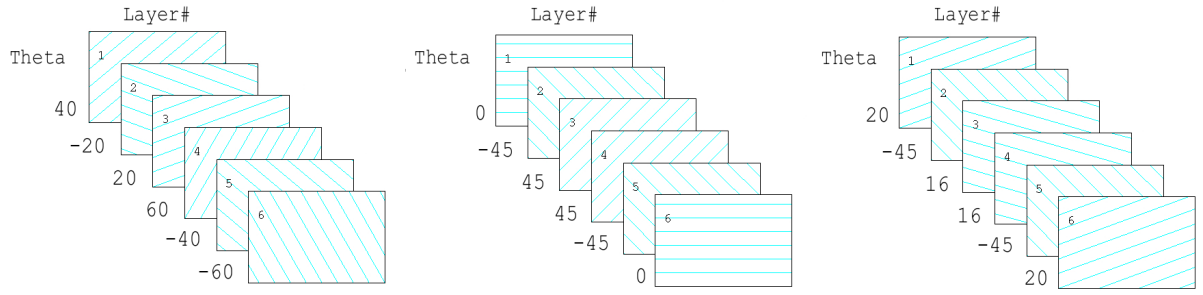
Check for coupling effects

The multi-layered element has been proved to pass the standard patch test for bending and membrane loading. This subsection complements the study of reliability for the laminated finite element by setting three orthotropic multi-layered layups and checking for the activation of the three existing coupling stiffness described in section 2.2 in the ABD matrix description of stiffnesses in Classic Lamination Theory (CLT). Listed here again for the reader's convenience:

- Shear - Extension coupling stiffness. Also known as A_{16} and A_{26}.
- Bending - Twist coupling stiffness. Also known as D_{16} and D_{26}.

- Bending - Extension coupling stiffness. Also known as B_{ij}.

The upcoming subsections check for the effect of the coupling stiffness by setting up the FE model described in figure 7.4 (only regular and tilted mesh) using a 6 UD laminae layered element. In each of the cases the direction of each lamina fibers are differently oriented in order to achieve a laminated element in which the wanted coupling stiffness are not equal to zero (figure 7.7). Hence, by having the appropriate layup and applying the correct load, the coupling effect is activated and simulated by means of the linear 4-node shell element with MITC and EAS enhancements.



(a) Bend-Extension layup. (b) Bend-Twist layup. (c) Shear-Extension layup.

Figure 7.7. Three layups corresponding to each checking test.

The aforementioned layups are meshed with the linear 4-node shell element of this work and have the material attributes of an orthotropic lamina made of CFRP. These are $E_1 = 138.0[GPa]$, $E_2 = E_3 = 9.0[GPa]$, $G_{12} = G_{13} = 7.10[GPa]$, $G_{23} = 6.20[GPa]$ and $\nu_{12} = 0.3$.

Bending-Extension

Using the layup shown in figure 7.7 (a), the cantilever beam is pulled with a force of 10000N.

Table 7.3. Displacement values for the bend-extension case. Units used [m].

		Laminated 4 MITC & EAS		
Load case	**Mesh**	u_x	u_y	u_z
BEND - EXTENSION	A	$6.270 \cdot 10^{-5}$	$1.330 \cdot 10^{-5}$	$2.601 \cdot 10^{-3}$
	B	$6.429 \cdot 10^{-5}$	$-5.071 \cdot 10^{-5}$	$2.645 \cdot 10^{-3}$
		ANSYS shell 181		
BEND - EXTENSION	A	$6.270 \cdot 10^{-5}$	$1.330 \cdot 10^{-5}$	$2.602 \cdot 10^{-3}$
	B	$5.901 \cdot 10^{-5}$	$-2.107 \cdot 10^{-6}$	$1.730 \cdot 10^{-3}$

As seen in table 7.3 when the mesh is not distorted the element is capable of modeling the bending-extension. While u_x is of a 10^{-5} magnitude, its bending coupling makes the beam bend and deform u_z of a magnitude a hundred times bigger. This is an expected behaviour as the bending stiffness is much smaller than the in-plane stiffness for such thin structures. Once the mesh is distorted, as seen in section 7.1.1 a displacement in the y direction appears due to the geometry irregularity.

Bending-Twist

Using the layup shown in figure 7.7 (b), a vertical a force of 10000N is applied at the tip of the cantilever beam.

Table 7.4. Displacement values for the bend-twist case. Units used [m].

		Laminated 4 MITC & EAS			
Load case	**Mesh**	u_x	u_y	u_z (node 2)	u_z (node 8)
BEND - TWIST	A	0.0	0.0	$2.366 \cdot 10^{-1}$	$2.382 \cdot 10^{-1}$
	B	0.0	0.0	$2.363 \cdot 10^{-1}$	$2.378 \cdot 10^{-1}$
		ANSYS shell 181			
BEND - TWIST	A	0.0	0.0	$2.365 \cdot 10^{-1}$	$2.382 \cdot 10^{-1}$
	B	0.0	0.0	1.118	1.134

Table 7.4 shows that when the beam undergoes a bending load the twist coupling is barely activated. The twisting stiffness exists in both meshes but does not appear as intense as the both the other cases.

To make sure the results are not a consequence of a wrongly implemented element, the same case has been simulated in ANSYS Mechanical APDL using shell 181. The ANSYS results seem to be more inaccurate when the mesh is distorted.

Shear-Extension

Using the layup shown in figure 7.7 (c), the cantilever beam is pulled with a force of 10000N.

Table 7.5. Displacement values for the shear-extension case. Units used [m].

		Laminated 4 MITC & EAS		
Load case	**Mesh**	u_x	u_y	u_z
SHEAR - EXTENSION	A	$3.220 \cdot 10^{-5}$	$-2.580 \cdot 10^{-6}$	0.0
	B	$3.300 \cdot 10^{-5}$	$-3.630 \cdot 10^{-5}$	0.0
		ANSYS shell 181		
SHEAR - EXTENSION	A	$3.850 \cdot 10^{-5}$	$-3.400 \cdot 10^{-6}$	0.0
	B	$3.220 \cdot 10^{-5}$	$-2.830 \cdot 10^{-6}$	0.0

The shear-extension results are shown in table 7.5. It is seen that while in an isotropic case, u_y is equal to zero, it has a value of the same magnitude as the displacement in the pulling direction when a suitable laminate is analyzed. This is due to the shear-extension coupling that appears in the formulation of laminate composite materials described in chapter 2. If the mesh

is distorted, the coupling is affected by the irregular deformation, shown in table 7.2, and the results of the coupling are not to trust if coupling related deformation are of interest.

7.1.3 Tests for general curved structures

Once the linear element has undergone the first basic benchmarks in a 2D flat shell structures, more general structures are modeled and analysed in order to make sure that the linear implementation in MUST is properly working for double curved shell structures. It is an important check because there is a lot of book-keeping and reused functionalities in MUST that have to be compatible with the 4-node shell element subroutine. The benchmarking examples are also taken from MacNeal and Harder (1985).

Scordelis-Lo roof

The first test is a single curved structure known as the Scordelis-Lo roof. This test proves whether the element can model and analyse single curved surfaces as a first step to a general double curved shell. The structure parameters and the boundary conditions can be found in figure 7.8, together with the meshed used.

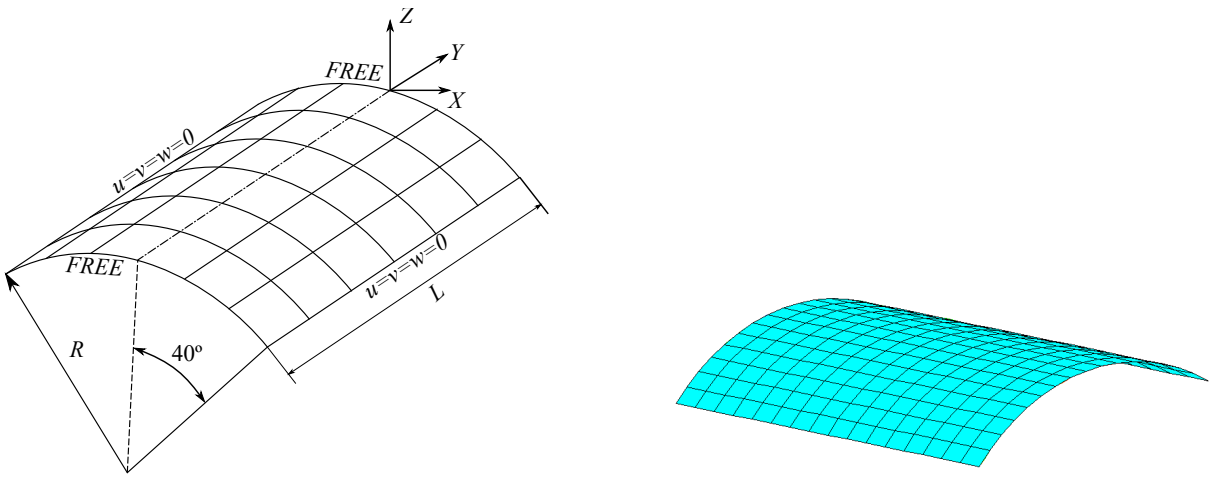


Figure 7.8. Scordelis-Lo roof where $R = 25m$, $thickness = 0.25m$, $E = 4.32x10^8 Pa$ and $\nu = 0$ with a constant pressure of $90Pa$. The mesh has 16x16 elements.

The results obtained in MUST are compared with the ones obtained in ANSYS with the shell 181 element, with full integration and incompatible modes keyoption activated, to asses the accuracy of the MUST implementation. Apart from the numerical result, shown in table 7.6, the displacement field distribution is also of interest and therefore is also displayed.

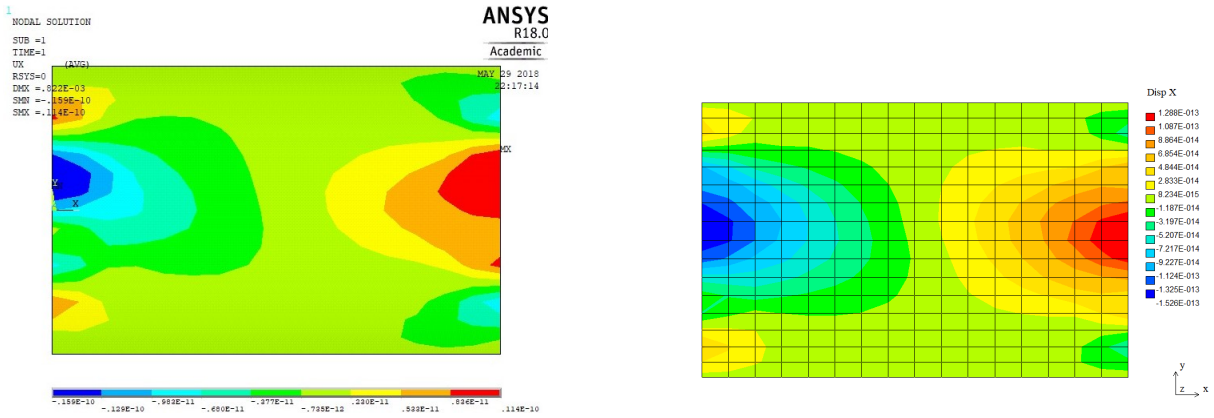


Figure 7.9. Displacement in the X direction. Left: ANSYS, Right: MUST.

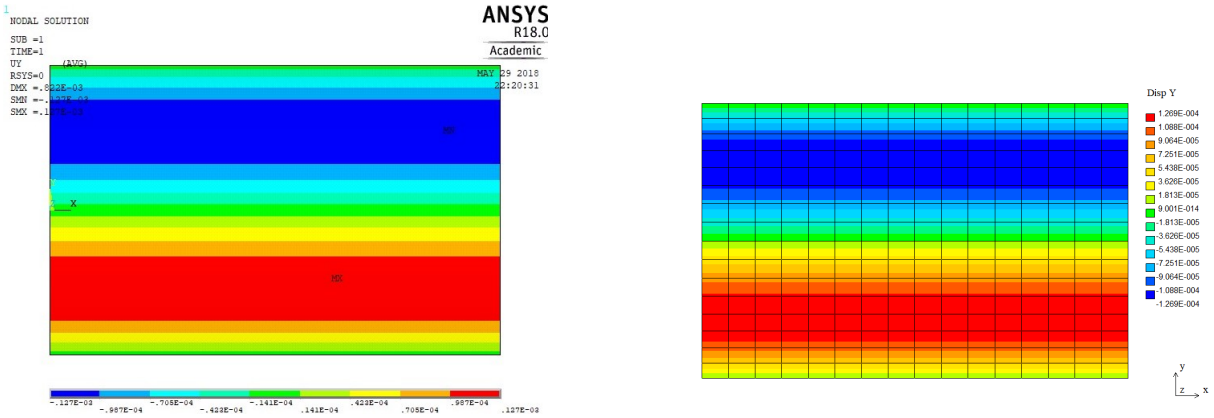


Figure 7.10. Displacement in the Y direction. Left: ANSYS, Right: MUST.

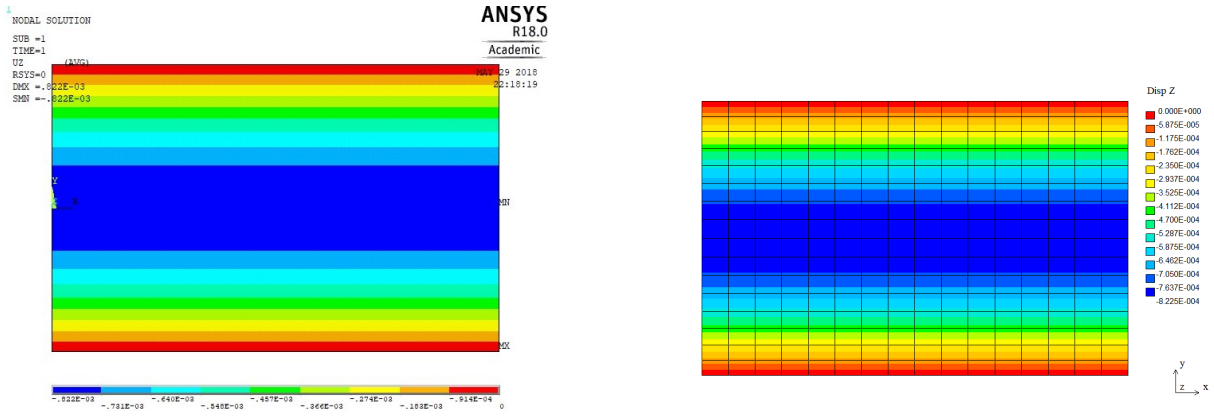


Figure 7.11. Displacement in the Z direction. Left: ANSYS, Right: MUST.

Table 7.6. Displacement results for the $Y = L$ point in the dashed line of the Scordelis-Lo roof. Results in meters.

Structure	**Element**	u_x	u_y	u_z
Scordelis-Lo roof	ANSYS shell 181	$-1.59 \cdot 10^{-11}$	$-1.23 \cdot 10^{-13}$	$-8.22 \cdot 10^{-4}$
	MUST 4-node	$-1.53 \cdot 10^{-13}$	$9.00 \cdot 10^{-14}$	$-8.23 \cdot 10^{-4}$

Hemisphere

The next benchmarking is the hemisphere with a 18° aperture on the top. A quarter of the structure is studied as seen in figure 7.12 to speed-up the analysis. This benchmark is one of the toughest benchmarks for shell elements and assesses whether the element is capable of model and analyse a general double curved shell. The results are summarized in table 7.7.

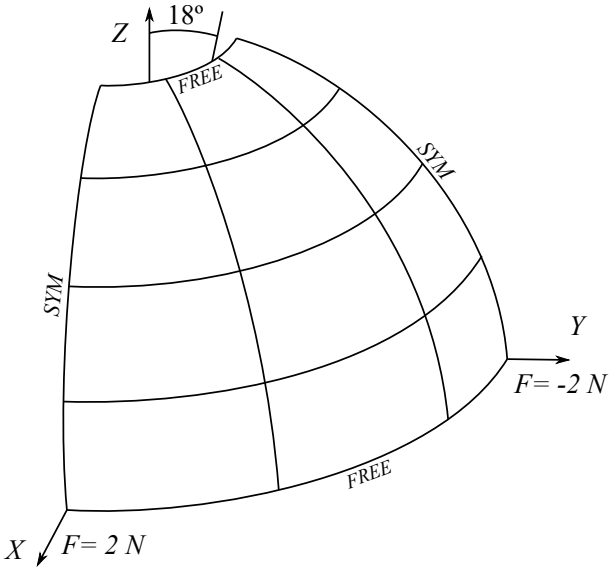


Figure 7.12. Hemisphere structure with the 18 hole on top. Only a quarter of the structure is studied. units in meters

As before, the results in MUST are compared with the ones in ANSYS, obtained using the shell 181 element with full integration and incompatible modes activated. The mesh used for these results is a fine mesh of $56x56$ elements.

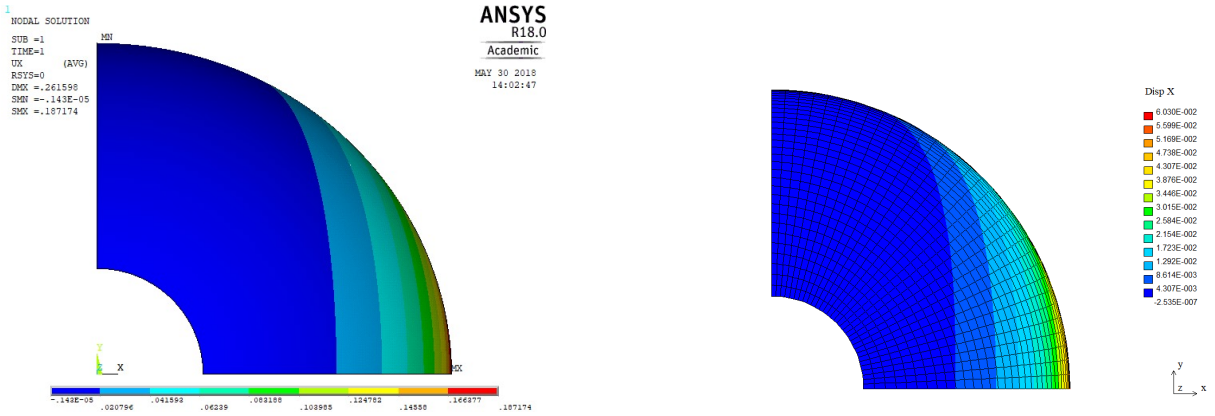


Figure 7.13. Displacement in the X direction. Left: ANSYS, Right: MUST.

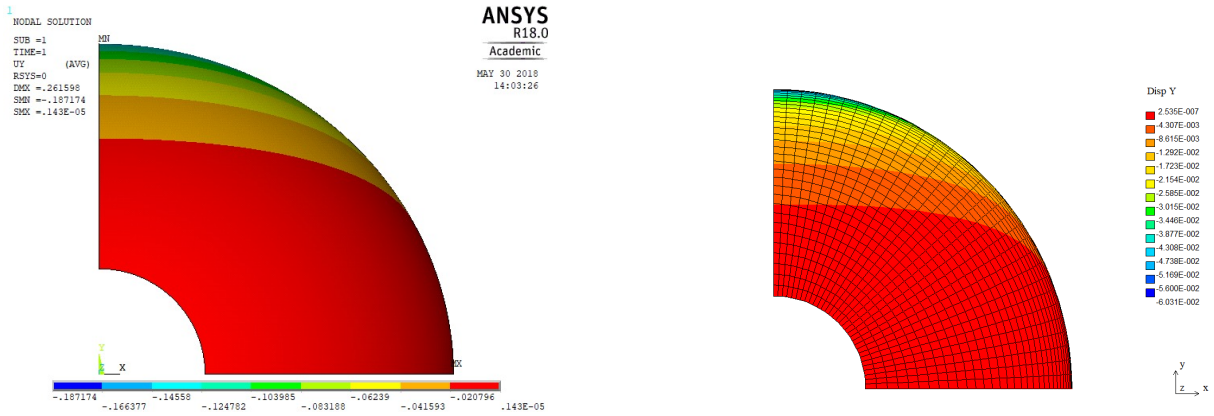


Figure 7.14. Displacement in the Y direction. Left: ANSYS, Right: MUST.

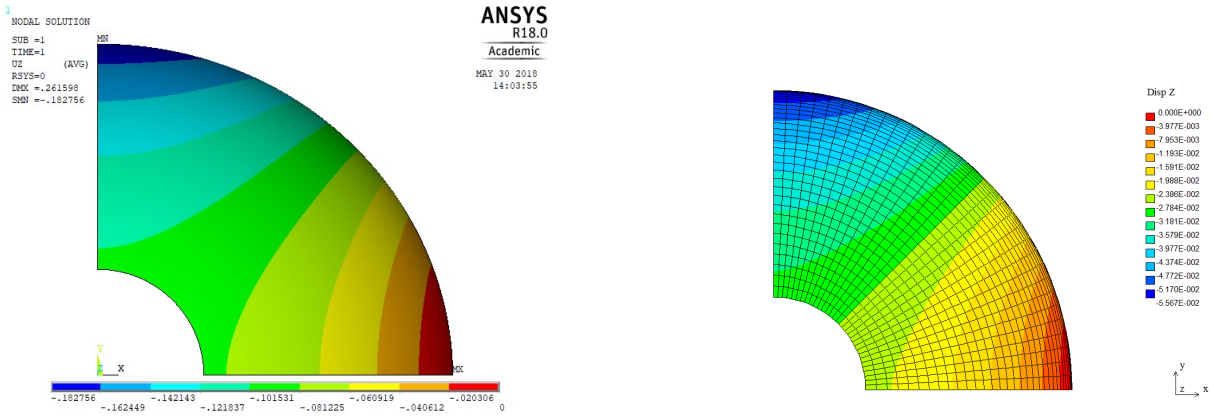


Figure 7.15. Displacement in the Z direction. Left: ANSYS, Right: MUST.

Table 7.7. Displacement results for the Hemisphere structure for the point on the top of the hemisphere coincident with the Y axis. The field with (*) was obtained with a fine mesh of 100x100 elements. Results in meters

Structure	**Element**	u_x	u_y	u_z
Hemisphere	ANSYS 181	0.00	$-1.89 \cdot 10^{-3}$	$-9.86 \cdot 10^{-1}$
	MUST 4-node(*)	0	$-1.75 \cdot 10^{-3}$	$-9.73 \cdot 10^{-1}$

The reason such a fine mesh is used in the results shown above is that convergence appeared to be really slow for the case of the quarter hemisphere. This can be observed in figure 7.16 where the convergence plot is presented. Although convergence is not achieved in the plot, with an extreme fine mesh of $140x140$ elements the results converge to a value of 0.182. This slow convergence rate is shared with the 9-node shell element already implemented in MUST.

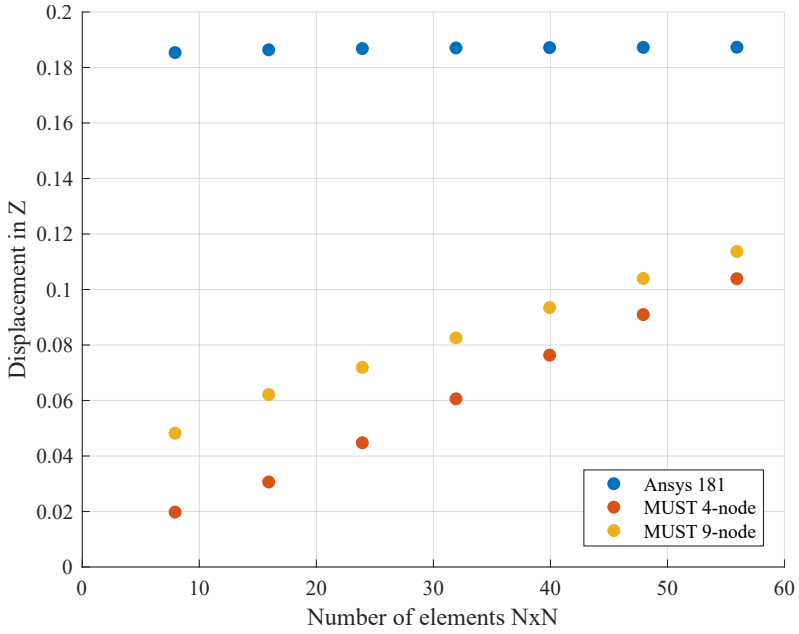


Figure 7.16. Displacement Z of the top of the hemisphere for different element types and meshes.

7.1.4 Linear element assessment

In this section a general assessment of the linear element is done in order to have full knowledge of its advantages and limitations in case of further use on complex shell laminated structures. Both the isotropic and laminated orthotropic applications are commented and assessed.

As a general comment, the linear 4-node shell element must be used with care (regardless the number of layers or application), both the one presented in this report and the shell 181 in ANSYS. For situations of out-of-plane bending the element yields good results with or without element distortion. However, in a situation of in-plane bending or extension the quality of the mesh shows a big impact on the results of the analysis.

In table 7.9 a compilation of the performance of the isotropic element coded in this project versus its equivalent in ANSYS is done. The sensitivity of the element performance is graded as follows:

Table 7.8. Scale of grades to evaluate the element performance.

Grade	Rule
Excellent	$2\% \geqslant error$
Good	$10\% \geqslant error > 2\%$
Poor	$20\% \geqslant error > 10\%$
Very Poor	$50\% \geqslant error > 20\%$
Fail	$error > 50\%$

Table 7.9. Assessment of the linear 4-node shell element and the ANSYS shell 181 in the load cases and meshes compared in the section.

Load case	Mesh	Linear Shell 4 MITC & EAS	ANSYS shell 181
Extension	A	Excellent	Excellent
	B	Excellent*	Very Poor*
	C	Excellent*	Good*
Twist	A	Good	Good
	B	Poor*	Poor*
	C	Very Poor*	Very Poor*
In-plane shear	A	Excellent	Excellent
	B	Very Poor	Very Poor
	C	Fail	Fail
Out-of-plane shear	A	Excellent	Excellent
	B	Good	Good
	C	Good	Good

Although the extensional displacement of the 4-node shell element with EAS is close to the theoretical value the element does not pass the assessment because of the behaviour explained in section 7.1.1.

As mentioned in subsection 7.1.2, the element is capable of modeling the bending-extension and shear-extension couplings properly. The particular case of bending-twist coupling appears in a very soft intensity. All the in-plane related couplings are sensitive to the mesh and display irregular behavior when the geometry of the element is distorted.

Finally, the similarities of the results obtained with the shell 181 element in ANSYS and the linear enhanced 4-node shell element with MITC and EAS, are due to both using incompatible modes and mixed interpolation methods. This is outlined in the introduction of Simo and Armero (1992), the paper in which the shell 181 ANSYS element is based on (Salvo and Swanson, 2017).

Regarding the benchmarking on the general shell structures the results are generally satisfactory. However the reason why the shell 181 element in ANSYS is able to converge so fast remains unclear. This difference in the convergence behaviour has only been noticed in the hemisphere example.

7.2 Linear buckling

The linear buckling has not yet been implemented in the MUST platform for the general 4-node shell element. However, a version of it for flat shells is up and running in Matlab. This version is used to compare the element behaviour to the linear buckling analysis in ANSYS. This result is also used to check the non-linear implementation of the element in section 7.3.

The structure analysed in this benchmarking is a rectangular plate of dimensions 6x3m with a thickness of 0.1m. The plate is meshed with 50x25 squared 4-node elements. The load case is that of a one side clamped plate (short side) and a unitary pressure applied in the other edge in a compressive direction. The material used is isotropic with $E = 10 \cdot 10^6 Pa$ and $\nu = 0.3$. The first four buckling modes corresponding to the buckling loads in table 7.10 are shown in Matlab and in ANSYS.

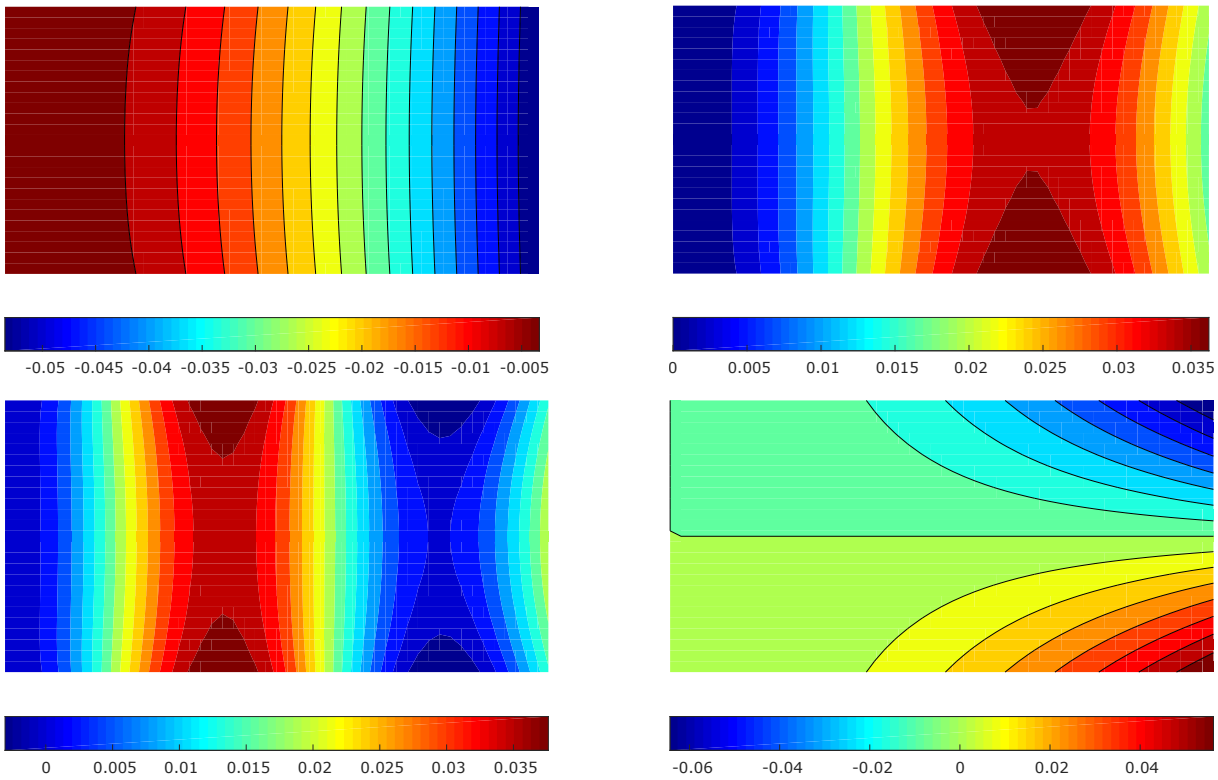


Figure 7.17. First 4 buckling modes in Matlab.

Table 7.10. First 4 buckling modes

Element/Mode	1	2	3	4
ANSYS 181	1.1828	10.681	30.031	33.388
Matlab 4-node	1.1827	10,675	29,987	33,352

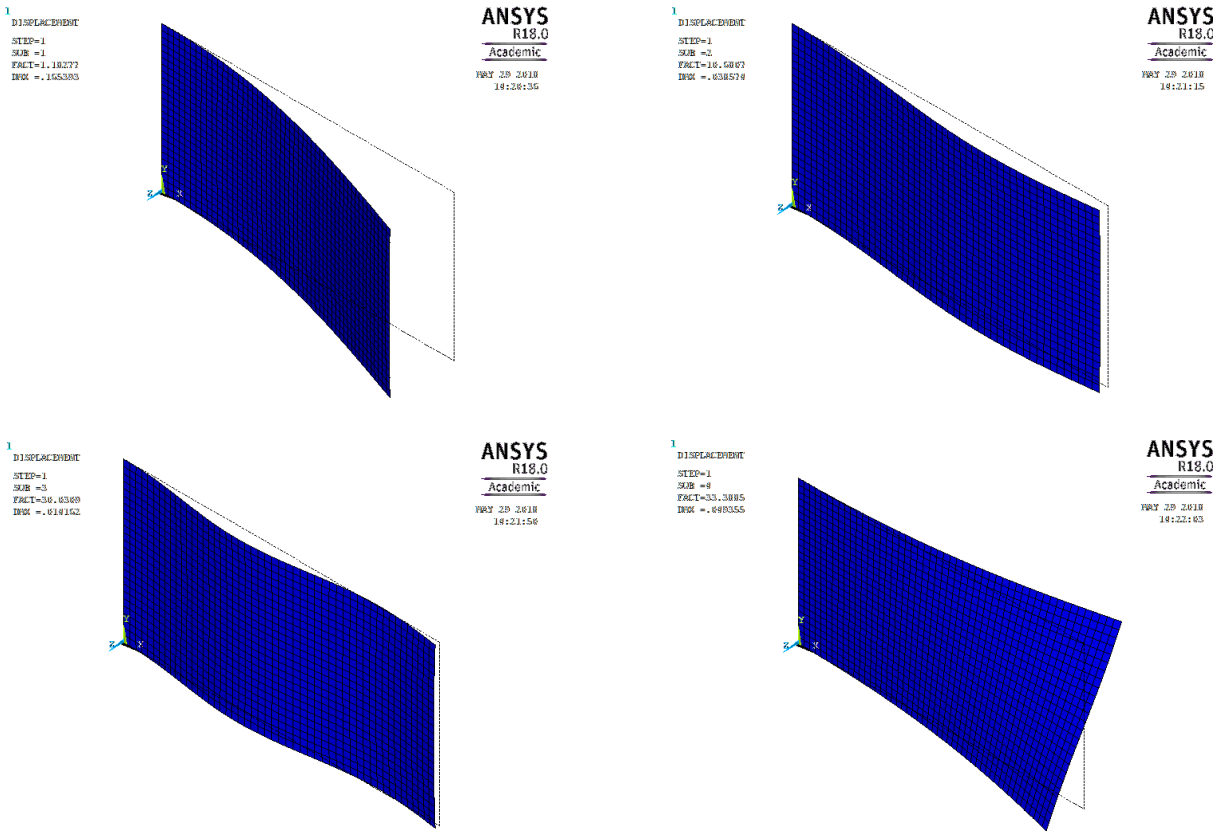


Figure 7.18. First 4 buckling modes in ANSYS.

As can be seen in table 7.10, excellent results are achieved in the linear buckling for the 4-node shell element in MATLAB. The results are expected to be the same in MUST as the same formulation will be used. These good results are also a prove of the correctness of the stress stiffness $[K_\sigma]$ matrix formulated in chapter 6.

7.3 Non-linear benchmarking

The benchmarking of the non-linear element is determined by the state of the non-linear implementation in MUST. At the date of delivery of this report, the non-linear implementation only works for flat shell structures that can be oriented in any direction of the space. As a consequence only one load step can be applied, because when the structure is updated with the deformed shape for the second or further load-step usually looses its flatness.

Thus, in this section the focus is put on checking that the flat implementation is working, and whether it provides similar results than the 9-node shell element in MUST. To prove that the non-linear implementation works, the buckling load found in the previous section will be applied to the same plate but meshed with non-linear 4-node and 9-node shell elements.

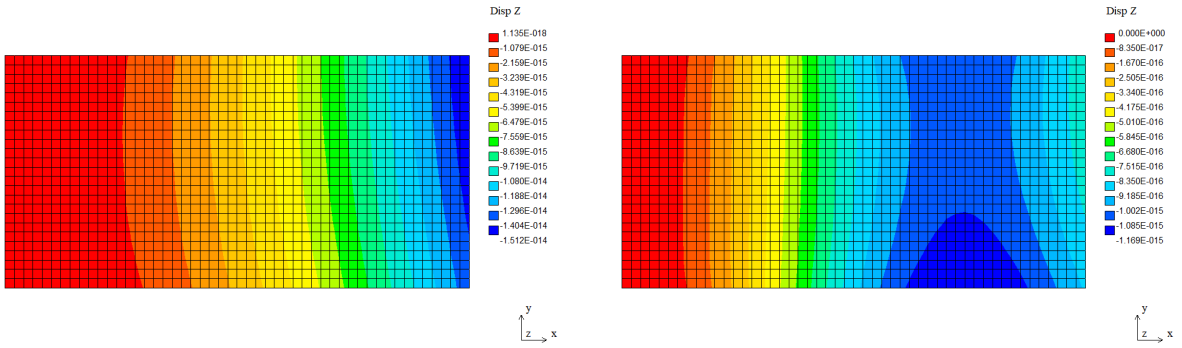


Figure 7.19. 1^{st} buckling mode through non-linear analysis without eccentricity. Left: 4-node shell. Right: 9-node shell.

As explained in chapter 6, non-linear analysis is often used to asses the buckling phenomena in detail as it is an inherent non-linear phenomenon. In figure 7.19 the buckling mode starts appearing, although it does so with small values of displacement due to the fact that no eccentricity in the load nor material imperfection has been applied. For the case of the 9-node shell, it even seems that the second mode is contributing to the buckling.

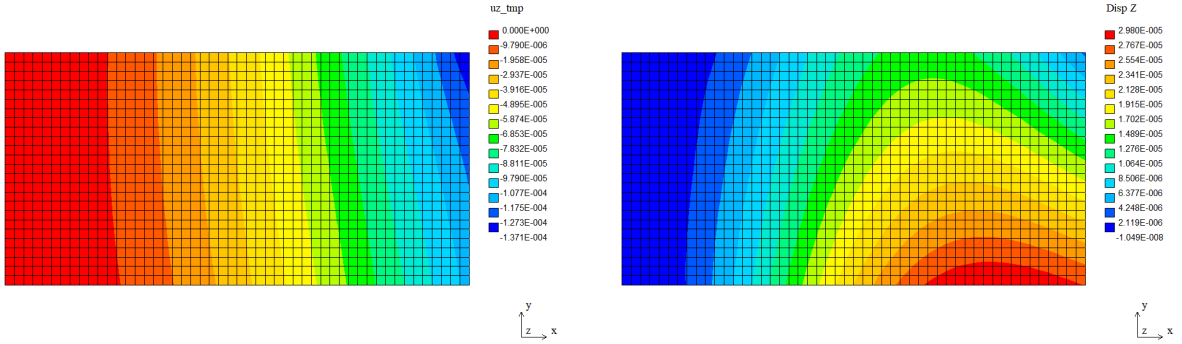


Figure 7.20. 1^{st} buckling mode through non-linear analysis with a load eccentricity. Left: 4-node shell. Right: 9-node shell.

In the second case, a small force of $F_z = 10^{-6}N$ is applied at each corner node of the right edge of the plate. In figure 7.20 it can be seen that now the buckling mode has been activated with higher displacements. In the case of the 9-node element, the analysis converge and yields the result in the figure while, for the 4-node element, the analysis needs to be manually terminated because instability makes it non convergent. The figure shows the last iteration before the residual starts growing. It can be seen that the plate is starting to tilt towards the deformation showed by the 9-node analysis.

The last test regarding non-linear effects and buckling is a simplification of the spar cap panel presented in Lautstsen and Vestergaard (2010). The spar cap panel is a structural part placed where the shear web meets the shell of a wind turbine blade. As the element formulated in the project is intended to be used in the structural optimization process of wind turbine blades, a scarp panel would have been a good benchmark example. However, the inability of the non-linear element currently implemented in MUST to properly model curved parts, the scarp panel is assumed to be flat. Thus, the same plate as before is used but the two long edges are fixed in the y and z directions, the right edge only in the z direction and the left edge in x and

z directions. The buckling load corresponding to the first mode, calculated with ANSYS as 2821.6N, is distributed along the right edge.

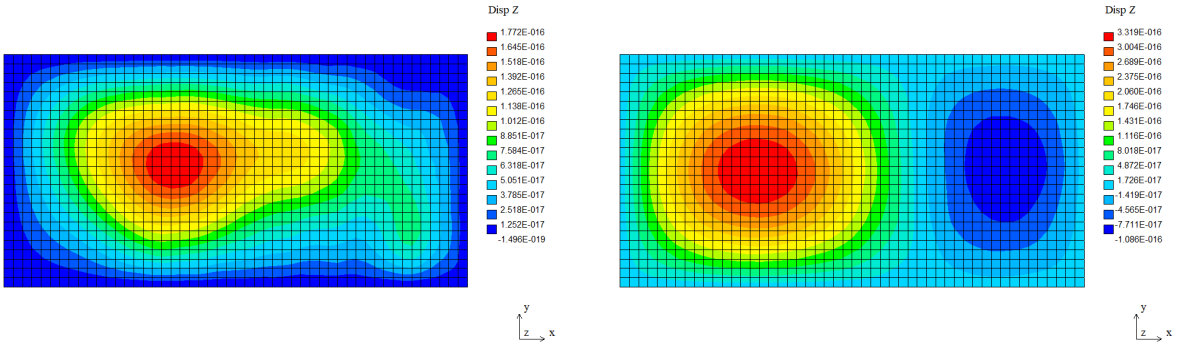


Figure 7.21. 1^{st} buckling mode through non-linear analysis of the simplified spar cap panel. Left: 4-node element. Right: 9-node element

As can be seen in figure 7.21, the 4-node element shows a less defined buckling mode than the 9-node. As before, the fact that there is no eccentricity applied makes the displacements in the z direction to be small.

As a conclusion for the non-linear analysis, the 4-node non-linear element shows to be working for flat cases. More work needs to be done to make sure that the implementation in MUST works for general double curved shell structures and for more than one loadstep.

Conclusions 8

The results shown in this work can be clearly separated in two parts in which different degrees of success have been achieved. Regarding the theoretical base, the knowledge acquired during the non-linear formulation of the element has been shown and explained in this report. The aim of the compilation of all the theories and derivations found in Klinkel et al. (1999), Bathe (1996), Washizu (1975), Simo and Rifai (1990) and Lund (2016a) was to be able to have a full fluid text with all the knowledge required to formulate such element. This compilation of different topics was difficult to find by the authors as a single text.

The numerical implementation of the theoretical concepts allows the user to get an insight of the element and give all the tools needed to code it. Regarding the implementation of the non-linear element in MUST, some extra work is still required as there is a bug in the code that does not allow the correct non-linear analysis of general double curved shells. In addition the code does not converge when more than one loadstep is applied. Despite that, the authors are sure that the theoretical formulation and the numerical implementation are correct as they use techniques that have been proven successful in other element formulations. The linear buckling analysis implemented in MATLAB and soon to be in MUST, provide a prove that the stress stiffness matrix (probably the most complex of this work) is well formulated. Regarding the linear element, the implementation is completely finished and giving accurate results compared to other shell elements as seen in the benchmarking of this report.

Regarding the linear and partially non-linear benchmarking, the element shows a good behavior that is sometimes better than the similar shell 181 element in ANSYS. The non-linear benchmarking for flat shells done through the buckling modes prove that the non-linear element can show good results too. Although it has not been extensively tested, the computational power required for the analyses with the 4-node shell element are lower than the 9-node already implemented while the results are similar, as has been observed at the benchmarking results. This was one of the main objectives of the project as this element is intended to be used for optimization of shell structures.

The next steps that need to be done if a faster and completely reliable element is desired apart from finishing the non-linear and the linear buckling implementation in MUST are the optimization of some parts of the code as well as an extensive benchmarking on the stress and strain post-processing. Overall, the authors of this thesis are satisfied of the knowledge gained during the development of this project and hope that the implementation in the MUST platform will be soon up and running.

References

Andelfinger, U. and E. Ramm (1993). Eas-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalent hr-elements. *International Journal for Numerical Methods in Engineering 36*, 1311–1337.

Bathe, K. J. (1996). *Finite Element Procedures.* Prentice Hall.

Bathe, K. J. and F. Brezzi (1987). A simplified analysis of two plate bending elements - the mitc4 and mitc9 elements . *Numerical Techniques for Engineering Analysis and Design*, 406–417.

Bathe, K. J., M. L. Bucalem, and F. Brezzi (1990). Displacement and stress convergence of our mitc plate bending elements. *Engineering Computations 7*(4), 291–302.

Bonet, J. and R. D. Wood (1997). *Nonlinear Continuum Mechanics for FInite Element Nalaysis.* Cambridge University Press.

Boothby, W. M. (1975). *An introduction to differentiablemanifolds and Riemann geometry.* Academic Press.

Cook, R. D., D. S. Malkus, M. E. Plesha, and R. J. Witt (2001). *Concepts and applications of finite element analysis (4th edition).* John Wiley & sons.

Cowper, G. R. (1966). The Shear Coefficient in Timoshenko's Beam Theory. *Journal of Applied Mechanics 2*(33), 335–340.

Crisfield, M. (2000). *Non-linear Finite Element Analysis of Solids and Structures.* John Wiley & sons.

Davis, H. F. and A. D. Snider (1979). *Introduction to vector analysis.* Allyn & Bacon Inc.

Dvorkin, E. N. and K. J. Bathe (1984). A continuum mechanics based four-node shell element for general non-linear analysis. *Engineering Computations 1*, 77–78.

Dvorkin, E. N. and K. J. Bathe (1985). A Four-node Plate Bending Element Based on Mindlin-Reissner Plate Plate heory and a Mixed Interpolation. *International Journal for Numerical Methods in Engineering 21*, 367–383.

Dym, C. L. and I. H. H. Shames (2013). *Solid Mechanics.* Springer.

Fleisch, D. (2012). *A student's Guide to Vectors and Tensors.* Cambridge University Press.

Flügge, W. (1972). *Tensor Analysis and Continuum Mechanics.* Springer-Verlag.

Greve, T., M. Ravn, and S. W. Torgard (2008). Improved Formulation of ESL Shell Elements. *Department of Mechanical Engineering, Aalborg Univsersity*.

Hansen, H. F. and C. G. Hvejsel (2007). Efficient Finite Element Formulation for Analysis and Optimization of Laminated Composite Shell Structures. *Department of Mechanical Engineering, Aalborg Univsersity*.

Heinbockel, J. H. (1996). *Introduction to Tensor Calculus and Continuum Mechanics*. Old Dominion Universityl.

Hughes, T. J. R. (1987). *The Finite Element Method*. Prentice-Hall.

Jensen, L. R., J. C. Rauhe, and J. Stegmann (2001). Finite elements for geometric non-linear analysis of composite laminates and sandwich structures. *M.Sc. Thesis, Department of Mechanical Engineering, Aalborg Univsersity*.

Johansen, L. and E. Lund (2009). Optimization of laminated composite structures using delamination criteria and hierarchical models. *Structural Multidisciplinary Optimization 38*, 358–375.

Johansen, L. S. (2009). *Analysis and Optimization of Composite Structures using Adaptative Analysis Methods. Ph. D. thesis*. Aalborg University.

Jones, R. (1998). *Mechanics Of Composite Materials*. Materials Science and Engineering Series. Taylor & Francis.

Kühnel, M. and T. Kraus (2016). *The global CFRP market*. Carbon Composites.

Klinkel, S., F. Gruttmann, and W. Wagner (1999). A continuum based three-dimensional shell element for laminated structures. *Computers and Strucutres 71*, 46–62.

Klinkel, S. and W. Wagner (1997). A Geometrical Non-Linear Brick Element Based on the EAS-Method. *International Journal for Numerical Methods in Engineering 40*, 4529–4545.

Ko, Y., P.-S. Lee, and K.-J. Bather (2016). The MITC4+ shell element and its performance. *Computers and Structures 169*, 57–68.

Lass, H. (1950). *Vector and Tensor Analysis*. McGraw-Hill.

Lautstsen, S. and J. Vestergaard (2010). Design study of a wind turbine blade. Master's thesis, Aalborg University, Aalborg, Denmark.

Lund, E. (2016a). *Analysis of Laminated Composite Structures*. Department of Mechanical and Manufacturing Engineering, Aalborg University.

Lund, E. (2016b). *Gradient Based Optimization of Laminated Composite Structures*. AAU Department of Mechanical and Manufacturing Engineering.

Lund, E. and E. Lindgaard (2016). *Notes for course on Finite Elements Method*. AAU Department of Mechanical and Manufacturing Engineering.

MacNeal, R. H. and R. L. Harder (1985). A Proposed Standard set of Problems to test Finite Element Accuracy. *Finite Elements in Analysis and Design 1*, 3–20.

Mazumdar, S. (2004). Composites growth realizing its global potential. `https://www.compositesworld.com/columns/composites-growth-realizing-its-global-potential`. Accessed: 27-05-2018.

Olhoff, N. and E. Lund (1995). Finite elements based engineering design sensitivity analysis and optimization. *Advances in Structural Optimization 1*, 1–45.

Ravn, M., S. W. Torgard, and T. Greve (2008). *Improved Formulation of ESL Shell Elements. MSc thesis.* Aalborg University.

Reddy, J. (2009). *Mechanics of laminated composite plates and shells. Theory and analysis.* CRC Press.

Rudnicki, J. W. (2014). *Fundamentals of Continuum Mechanics.* Wiley.

Saad, Y. (1992). *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press.

Salvo, G. J. D. and J. A. Swanson (2017). *ANSYS Engineering Analysis System User's Manual (18.2 release.* Swanson Analysis Systems.

Simo, J. and F. Armero (1992). Geometrically Non-Linear Enhanced Strain Mixed Methods and the Method of Incompatible Modes. *International Journal for Numerical Methods in Engineering 33*, 1413–1449.

Simo, J. C. and S. Rifai (1990). A class of mixed assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering 29*, 1595–1638.

Sloan, J. (2016). Survey: Composites industry trends. market survey.

Smith, D. E. and M. L. Latham (1925). *The geometry of Rene Descartes.* Open Court Publishing Co.

Stegmann, J. (2005). Analysis and Optimization of Laminated Composite Shell Structures. *Ph.D Thesis, Department of Mechanical Engineering, Aalborg Univsersity.*

Synge, J. L. and A. Schild (1978). *Tensor Calculus.* Dover Publications.

Taylor, R. L., J. C. Simo, O. Zienkiewicz, and A. C. H. Chan (1986). The Patch Test - A condition for assessing FEM convergence. *International Journal for Numerical Methods in Engineering 22*, 39–62.

Timoshenko, S. and S. Woinowsky-Krieger (1987). *Theory of Plates and Shells.* McGraw-Hill Book Company, Inc.

Timoshenko, S. P. (1922). On the transverse vibrations of bars of uniform cross-section. *Philosophical Magazine* (43), 125–131.

Vlachoutsis, S. (1992). Shear Correction Factors for Plates and Shells. *International Journal for Numerical Methods in Engineering 33*, 1537–1552.

Washizu, K. (1975). *Variational Methods in Elasticity and Plasticity.* Pergamon Press.

Wilson, E. L., R. L. Taylor, W. P. DOherty, and J. Ghaboussi (1973). Incompatible displacement models. *Numerical and Computer Models in Structural Mechanics*, 43–57.

Basic linear FEM concepts A

Finite Element Methods. Discrete formulation.

In a general sense, the FEM is a very versatile and applied numerical method used to solve complex partial differential equations that describe physical problems. In structural mechanics, the equations to be solved comprise the elastic problem formulated by means of calculus of variations (Dym and Shames, 2013).

Discretization and interpolation

The core of approximating the equations from theory of elasticity relies on discretizing a continuum media with piece-wise continuous elements. To do so, a real problem is needed. Then a mathematical model is made. Last, the mathematical model is discretized by dividing it into a mesh of finite elements that are able to represent a continuous field as a piece-wise approximation. Figure A.1 gives a graphical explanation of the process from having a real structure, a wind turbine blade, to reaching its appropriate finite element model by means of discretization with shell elements. As it is seen in the figure, the shell model of the wind turbine is divided into a series of simple elements. Each element is a special domain defined by a number of nodes from which any state variable value is interpolated across the domain of the element. In the case of the 4-node shell element developed in this project, such functions are mapped into a natural element space (r, s, t) that among other nice properties make the element remain bilinear regardless its physical shape or orientation (Cook et al., 2001).

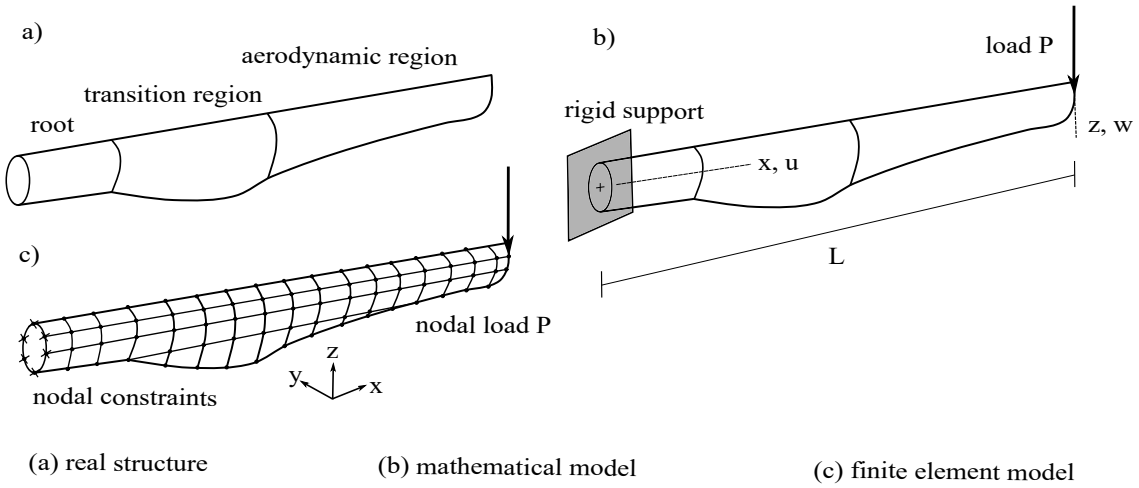


Figure A.1. Wind turbine shell modelling and discretization.

The principle of stationary potential energy

The principle of stationary potential energy is used to derive the governing equations for the static elastic problem. It is a variational approach based on integral expressions describing a problem: a functional. Theoretical background of calculus of variations is found in Dym and Shames (2013). In the case of FEM, the functional used is the total potential energy of an elastic body Π, which consists of the internal potential energy Π_i stored as strain energy and the external potential energy Π_e representing the work done by external forces:

$$\Pi = \Pi_i + \Pi_e \tag{A.1}$$

Deeply explained in Dym and Shames (2013), the principle of stationary potential energy states that "*an admissible displacement field being related through a constitutive relation to a stress field in equilibrium must extremize the total potential energy of a discrete system wrt. all other kinematically admissible displacements*":

$$\frac{\partial \Pi_{app}}{\partial \{D\}} = \{0\} \tag{A.2}$$

The functional described in equation (A.1) is an integral expression of continuous functions. In order to discretize it the Rayleigh-Ritz method is applied. Basically, it consists of substituting the partial differential equations by algebraic equations that approximate the fields by means of polynomials, also called interpolation functions or shape functions. The result is a substitute model described by algebraic equations with a finite number of degrees of freedom (d.o.f) capable of yielding the exact solution if the exact field is included in the approximated field.

The basis of displacement based Finite Element Analysis

(Note: in this section all element quantities are expressed in the global reference system.)

Being the principle of stationary potential energy presented, the basis of displacement based FEA is inserting the interpolated nodal displacements into the expression of potential energy:

$$\Pi_{app} = \Pi_i^{app} + \Pi_e^{app} = \frac{1}{2}\sum_{i=1}^{n_e}\{u_e\}^T[k]\{u_e\} - \sum_{i=1}^{n_e}\{u_e\}^T\{r_e\} - \{d\}^T\{f_{ext}\} \tag{A.3}$$

where $[k]$, $\{u_e\}$, $\{r_e\}$ and $\{f_{ext}\}$ are the element stiffness matrix, the nodal displacement vector, the consistent load vector and the external force vector respectively.

Note from equation (A.3) that the element stiffness matrix $[k]$ is defined as

$$[k]_e = \int_{V_e}[B]^T[C][B]dV \tag{A.4}$$

and the consistent load vector $\{r_e\}$ is:

$$\{r_e\} = \int_{V_e}[N]^T\{b\}dV + \int_{S_e}[N]^T\{\Phi\}dS \tag{A.5}$$

being $\{b\}$ the body forces and $\{\Phi\}$ the surface tractions.

When writing the approximate strains $\{\varepsilon_{app}\}$ as a function of the approximate displacements $\{u_{app}\}$ the generalized strain vector is written as

$$\{\varepsilon_{app}\} = [\partial]\{u_{app}\} = [\partial][N]\{u_e\} \equiv [B_0]\{u_e\} \tag{A.6}$$

where $[B]$ is the strain-displacement matrix, which in a nutshell, contains the derivatives of the displacement interpolations.

Now, the global stiffness matrix $[K]$ is introduced as well as a the global consistent load vector $\{R\}$ comprising the volume forces, surface tractions and external forces:

$$[K] = \sum_{j=1}^{n_e} [k_e], \qquad \{R\} = [N]^T\{b\} + [N]^T\{\Phi\} - \{F_{ext}\}$$

Eventually, the total potential energy of a discrete elastic body can be written as follows:

$$\Pi_{app} = \frac{1}{2}\{D\}^T[K]\{D\} - \{D\}^T\{R\} \tag{A.7}$$

Once the total potential is defined as a functional of the displacement fields approximated by the finite elements, requiring stationarity of the potential energy gives the governing equations of the problem:

$$\frac{\partial \Pi_{app}}{\partial \{D\}} = [K]\{D\} - \{R\} = \{0\} \tag{A.8}$$

Equation (A.8) comprises the algebraic equations to solve when dealing with linear, elastic, static problems. For deeper knowledge of the derivation of the FEM governing equations for linear and non-linear analysis check Cook et al. (2001) or Lund and Lindgaard (2016).

Outline of the isoparametric formulation

Section 6.1 starts with the mapping of the 3 dimensions (x, y, z) of the shell element in auxiliary coordinates, also called natural coordinates (r, s, t). Natural coordinates map the physical element into a reference element that is a cube. This is the principle of the so called isoparametric formulation.

This section, however, presents the isoparametric formulation through an easier element, the 2D 4-node isoparametric plane element (Lund and Lindgaard, 2016, lec.2) and (Cook et al., 2001, chap.4).

The isoparametric formulation makes it possible to define compatible elements that are non-rectangular and have curved edges. The methodology consists in formulating the element in the convective natural coordinates (r, s) where it has axis-parallel element edges, which make it compatible. Such element formulation uses the same shape functions for both geometric and kinematic variables. For the basic 2D isoparametric element as well as for the shell element formulation presented in the next chapter, the in-plane shape functions that are used are defined as follows:

$$N_1 = \frac{1}{4}(1-r)(1-s) \quad N_2 = \frac{1}{4}(1+r)(1-s) \quad N_3 = \frac{1}{4}(1+r)(1+s) \quad N_4 = \frac{1}{4}(1-r)(1+s)$$

The most remarkable features of using isoparametric formulation are the fact of having a compatible element regardless a distorted geometry and the ease of use when integrating by means of Gauss quadrature. The reason that makes it possible to use Gauss integration rules is the element length always going from $[-1\ ,1]$, when mapped in its natural coordinates. In addition the integration is exact when the Jacobian is constant through the element.

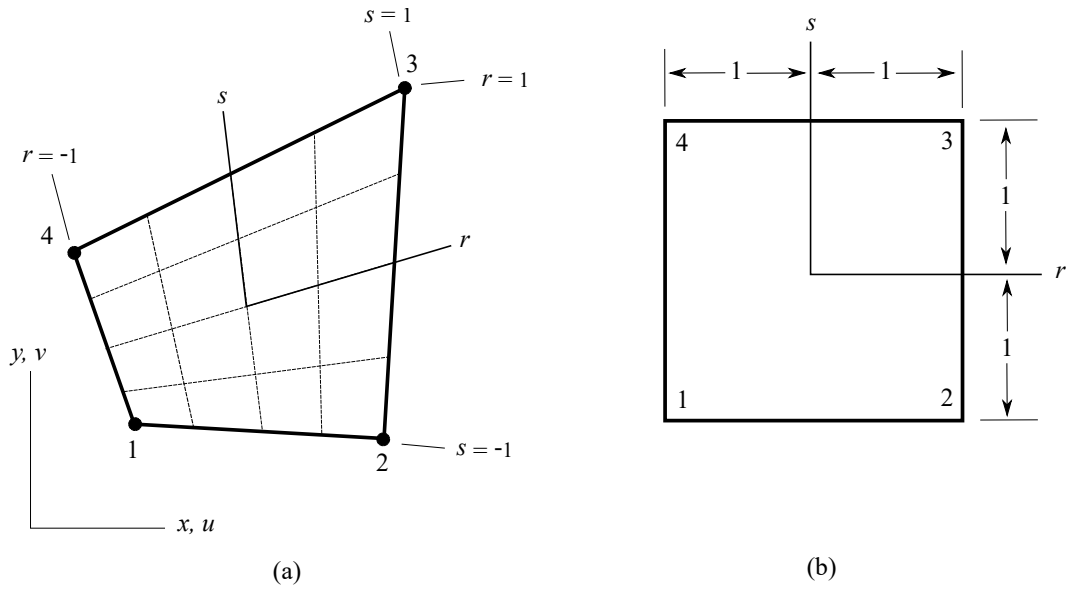


Figure A.2. (a) 4-node plane element in physical space. (b) The same element mapped into (r, s) space.

As it is thoroughly explained in the report, the governing equations are solved using the global coordinate system. Thus, when formulating the element stiffness matrix $[k]$, the strain-displacement matrix $[B_0]$ requires a transformation from the natural mapping to the global system.

The goal is to compute the strains in the global Cartesian coordinates as shown in equation (A.9).

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{Bmatrix} \tag{A.9}$$

In order to get the displacement field's derivatives with respect to the global coordinates (x, y) a transformation is needed because the displacement interpolation is a function of the natural coordinates (r, s).

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{Bmatrix} = \left[\Gamma_{exp}\right] \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \end{Bmatrix} \tag{A.10}$$

Therefore the derivation ahead is meant to find such relation by means of using the chain rule of differentiation. If only the derivatives of one direction (u) are taken, the following is obtained:

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{Bmatrix} \tag{A.11}$$

However, derivatives with respect to x and y are not available directly. Because of that we must

begin with its inverse

$$\begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \Big[J\Big] \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} \tag{A.12}$$

where $[J]$ is the Jacobian matrix, which gives the relationship between the two coordinate systems. By performing the inverse to the Jacobian matrix, the desired transformation is achieved:

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{Bmatrix} \quad \text{where} \quad [\Gamma] = [J]^{-1} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \tag{A.13}$$

in which $|J|$ is an area scale factor that relates the areas $\partial x \partial y$ and $\partial r \partial s$.

Eventually, in the case of the 2D 4-node isoparametric plane element the proper transformation is then

$$\Big[\Gamma_{exp}\Big] = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & 0 & 0 \\ \Gamma_{21} & \Gamma_{22} & 0 & 0 \\ 0 & 0 & \Gamma_{11} & \Gamma_{12} \\ 0 & 0 & \Gamma_{21} & \Gamma_{22} \end{bmatrix} \tag{A.14}$$

From equation (A.10), the vectors of displacement derivatives with respect to the natural coordinates is calculated as follows:

$$\begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \end{Bmatrix} = \Big[Q\Big] \Big\{u\Big\}_e = \begin{bmatrix} \frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} & 0 \\ \frac{\partial N_1}{\partial s} & 0 & \frac{\partial N_2}{\partial s} & 0 & \frac{\partial N_3}{\partial s} & 0 & \frac{\partial N_4}{\partial s} & 0 \\ 0 & \frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} \\ 0 & \frac{\partial N_1}{\partial s} & 0 & \frac{\partial N_2}{\partial s} & 0 & \frac{\partial N_3}{\partial s} & 0 & \frac{\partial N_4}{\partial s} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \tag{A.15}$$

Leading to a $[Q]$ matrix able to transform from the natural mapping to the global coordinate system.

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \Big[Q\Big] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \tag{A.16}$$

where

$$[Q] = \left[\Gamma_{exp}\right] \begin{bmatrix} \frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} & 0 \\ \frac{\partial N_1}{\partial s} & 0 & \frac{\partial N_2}{\partial s} & 0 & \frac{\partial N_3}{\partial s} & 0 & \frac{\partial N_4}{\partial s} & 0 \\ 0 & \frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} \\ 0 & \frac{\partial N_1}{\partial s} & 0 & \frac{\partial N_2}{\partial s} & 0 & \frac{\partial N_3}{\partial s} & 0 & \frac{\partial N_4}{\partial s} \end{bmatrix} \tag{A.17}$$

Finally, $[B_0]$ is defined as:

$$[B_0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} [Q_{exp}] \tag{A.18}$$

MITC matrices B

$$[H_{MITC}]_I = \begin{bmatrix} 0 & 0 & (1-r) & (1-r) & 0 & 0 & (1+r) & (1+r) \\ (1-s) & (1-s) & 0 & 0 & (1+s) & 0 & (1+s) & 0 \end{bmatrix} \tag{B.1}$$

$$[Q_{MITC}]_I = \begin{bmatrix} \frac{dN_I}{dr}\big|_B & 0 & 0 & -\frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{21}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{11}^I \\ 0 & \frac{dN_I}{dr}\big|_B & 0 & -\frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{22}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{12}^I \\ 0 & 0 & \frac{dN_I}{dr}\big|_B & -\frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{23}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_B V_{13}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_B^I V_{21}^I & \frac{1}{2}thN_B^I V_{11}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_B^I V_{22}^I & \frac{1}{2}thN_B^I V_{12}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_B^I V_{23}^I & \frac{1}{2}thN_B^I V_{13}^I \\ \frac{dN_I}{dr}\big|_A & 0 & 0 & -\frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{21}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{11}^I \\ 0 & \frac{dN_I}{dr}\big|_A & 0 & -\frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{22}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{12}^I \\ 0 & 0 & \frac{dN_I}{dr}\big|_A & -\frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{23}^I & \frac{1}{2}th\frac{dN_I}{dr}\big|_A V_{13}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_A^I V_{21}^I & \frac{1}{2}thN_A^I V_{11}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_A^I V_{22}^I & \frac{1}{2}thN_A^I V_{12}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_A^I V_{23}^I & \frac{1}{2}thN_A^I V_{13}^I \\ \frac{dN_I}{dr}\big|_D & 0 & 0 & -\frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{21}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{11}^I \\ 0 & \frac{dN_I}{ds}\big|_D & 0 & -\frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{22}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{12}^I \\ 0 & 0 & \frac{dN_I}{ds}\big|_D & -\frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{23}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_D V_{13}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_D^I V_{21}^I & \frac{1}{2}thN_D^I V_{11}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_D^I V_{22}^I & \frac{1}{2}thN_D^I V_{12}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_D^I V_{23}^I & \frac{1}{2}thN_D^I V_{13}^I \\ \frac{dN_I}{ds}\big|_C & 0 & 0 & -\frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{21}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{11}^I \\ 0 & \frac{dN_I}{ds}\big|_C & 0 & -\frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{22}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{12}^I \\ 0 & 0 & \frac{dN_I}{ds}\big|_C & -\frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{23}^I & \frac{1}{2}th\frac{dN_I}{ds}\big|_C V_{13}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_C^I V_{21}^I & \frac{1}{2}thN_C^I V_{11}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_C^I V_{22}^I & \frac{1}{2}thN_C^I V_{12}^I \\ 0 & 0 & 0 & -\frac{1}{2}hN_C^I V_{23}^I & \frac{1}{2}thN_C^I V_{13}^I \end{bmatrix} \tag{B.2}$$

$$[G_{MITC}] = \begin{bmatrix} \{g_3^B\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} \\ \{0\} & \{g_1^B\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} \\ \{0\} & \{0\} & \{g_3^A\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} \\ \{0\} & \{0\} & \{0\} & \{g_2^A\} & \{0\} & \{0\} & \{0\} & \{0\} \\ \{0\} & \{0\} & \{0\} & \{0\} & \{g_3^D\} & \{0\} & \{0\} & \{0\} \\ \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{g_1^D\} & \{0\} & \{0\} \\ \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{g_3^C\} & \{0\} \\ \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{0\} & \{g_2^C\} \end{bmatrix} \tag{B.3}$$

$$[Q_{MITC}]_{expanded} = \left[[Q_{1MITC}] \; [Q_{2MITC}] \; [Q_{3MITC}] \; [Q_{4MITC}] \right] \tag{B.4}$$

Stress Stiffness matrices C

The membrane terms are defined as:

$$\hat{S}^m_{IJ} = N_{I,r}N_{J,r}S_{11} + N_{I,s}N_{J,s}S_{22} + \left(N_{I,r}N_{J,s} + N_{I,s}N_{J,r}\right)S_{12}$$

$$\begin{aligned}
\hat{A}^m_{IJ} = &\left(-\frac{1}{2}h_I N_{I,r}\sum_{i=1}^{3}\{V_{2i}\}\right)\left(-\frac{1}{2}h_J N_{J,r}\sum_{i=1}^{3}\{V_{2i}\}\right)S_{11} \\
&+\left(-\frac{t}{2}h_I N_{I,s}\sum_{i=1}^{3}\{V_{2i}\}\right)\left(-\frac{t}{2}h_J N_{J,s}\sum_{i=1}^{3}\{V_{2i}\}\right)S_{22} \\
&+\Bigg[\left(-\frac{t}{2}h_I N_{I,r}\sum_{i=1}^{3}\{V_{2i}\}\right)\left(-\frac{t}{2}h_J N_{J,s}\sum_{i=1}^{3}\{V_{2i}\}\right) \\
&+\left(-\frac{t}{2}h_I N_{I,s}\sum_{i=1}^{3}\{V_{2i}\}\right)\left(-\frac{t}{2}h_J N_{J,r}\sum_{i=1}^{3}\{V_{2i}\}\right)\Bigg]S_{12}
\end{aligned}$$

$$\begin{aligned}
\hat{B}^m_{IJ} = &\left(-\frac{1}{2}h_I N_{I,r}\sum_{i=1}^{3}\{V_{1i}\}\right)\left(-\frac{1}{2}h_J N_{J,r}\sum_{i=1}^{3}\{V_{1i}\}\right)S_{11} \\
&+\left(-\frac{t}{2}h_I N_{I,s}\sum_{i=1}^{3}\{V_{1i}\}\right)\left(-\frac{t}{2}h_J N_{J,s}\sum_{i=1}^{3}\{V_{1i}\}\right)S_{22} \\
&+\Bigg[\left(-\frac{t}{2}h_I N_{I,r}\sum_{i=1}^{3}\{V_{1i}\}\right)\left(-\frac{t}{2}h_J N_{J,s}\sum_{i=1}^{3}\{V_{1i}\}\right) \\
&+\left(-\frac{t}{2}h_I N_{I,s}\sum_{i=1}^{3}\{V_{1i}\}\right)\left(-\frac{t}{2}h_J N_{J,r}\sum_{i=1}^{3}\{V_{1i}\}\right)\Bigg]S_{12}
\end{aligned}$$

The enhanced terms are defined as:

$$\hat{S}^s_{IJ} = 0 \tag{C.1}$$

$$\hat{A}^s_{IJ} = \frac{1}{2}\left[(1-s)_\alpha \varepsilon^B_{IJ} + (1+s)_\alpha \varepsilon^D_{IJ}\right]S_{13} + \frac{1}{2}\left[(1-r)_\alpha \varepsilon^A_{IJ} + (1+r)_\alpha \varepsilon^C_{IJ}\right]S_{23} \tag{C.2}$$

$$\hat{B}^s_{IJ} = \frac{1}{2}\left[(1-s)_\beta \varepsilon^B_{IJ} + (1+s)_\beta \varepsilon^D_{IJ}\right]S_{13} + \frac{1}{2}\left[(1-r)_\beta \varepsilon^A_{IJ} + (1+r)_\beta \varepsilon^C_{IJ}\right]S_{23} \tag{C.3}$$

Expression (C.1) is a consequence of the displacement in the t direction not having an effect on the transverse strains. The strain terms in (C.2) and (C.3) are calculated as follows:

$$
\begin{aligned}
{}_{\alpha}\varepsilon_{IJ}^{A/C} &= \left(-\frac{t}{2}h_I N_{I,s}^{A/C}\sum_{i=1}^{3}V_{2i}^{I}\right)\left(-\frac{1}{2}h_J N_{J}^{A/C}\sum_{i=1}^{3}V_{2i}^{J}\right)+\left(-\frac{1}{2}h_I N_{I}^{A/C}\sum_{i=1}^{3}V_{2i}^{I}\right)\left(\frac{t}{2}h_J N_{J,s}^{A/C}\sum_{i=1}^{3}V_{2i}^{J}\right)\\
{}_{\beta}\varepsilon_{IJ}^{A/C} &= \left(-\frac{t}{2}h_I N_{I,s}^{A/C}\sum_{i=1}^{3}V_{1i}^{I}\right)\left(-\frac{1}{2}h_J N_{J}^{A/C}\sum_{i=1}^{3}V_{1i}^{J}\right)+\left(-\frac{1}{2}h_I N_{I}^{A/C}\sum_{i=1}^{3}V_{1i}^{I}\right)\left(\frac{t}{2}h_J N_{J,s}^{A/C}\sum_{i=1}^{3}V_{1i}^{J}\right)\\
{}_{\alpha}\varepsilon_{IJ}^{B/D} &= \left(-\frac{t}{2}h_I N_{I,r}^{B/D}\sum_{i=1}^{3}V_{2i}^{I}\right)\left(-\frac{1}{2}h_J N_{J}^{B/D}\sum_{i=1}^{3}V_{2i}^{J}\right)+\left(-\frac{1}{2}h_I N_{I}^{B/D}\sum_{i=1}^{3}V_{2i}^{I}\right)\left(\frac{t}{2}h_J N_{J,r}^{B/D}\sum_{i=1}^{3}V_{2i}^{J}\right)\\
{}_{\beta}\varepsilon_{IJ}^{B/D} &= \left(-\frac{t}{2}h_I N_{I,r}^{B/D}\sum_{i=1}^{3}V_{1i}^{I}\right)\left(-\frac{1}{2}h_J N_{J}^{B/D}\sum_{i=1}^{3}V_{1i}^{J}\right)+\left(-\frac{1}{2}h_I N_{I}^{B/D}\sum_{i=1}^{3}V_{1i}^{I}\right)\left(\frac{t}{2}h_J N_{J,r}^{B/D}\sum_{i=1}^{3}V_{1i}^{J}\right)
\end{aligned}
$$