



UNIVERSITAT POLITÈCNICA DE CATALUNYA

BARCELONATECH

Escola Superior d'Enginyeries Industrial,  
Aeroespacial i Audiovisual de Terrassa

# TREBALL DE FI DE MÀSTER

**TFM TITLE:** Large Deformation Study of a Wing

**DEGREE:** Màster en Enginyeria Aeronàutica

**AUTHOR:** Ignacio Izquierdo Pérez

**ADVISOR:** Àlex Ferrer Ferré

**DATE:** 25 de abril de 2018



**Title :** Large Deformation Study of a Wing

**Author:** Ignacio Izquierdo Pérez

**Advisor:** Àlex Ferrer Ferré

**Date:** April 25, 2018

## Overview

Wing loading test is one of the most important milestones a manufacturer has to pass during the aircraft certification process. In this context, finite element simulations can arise as an excellent tool to reduce time in design processes prior to test, as well as to serve on posteriori analysis. Due to utmost importance of wing loading test the structural model requires consideration of the nonlinear characteristics associated to the wing.

This work presents the basis of both finite element method and nonlinear continuum analysis to afterwards implement them in a collaborative software formed by other four students and called *FEM-MAT-OO*. First, the finite element method is introduced within the framework of linear elasticity. Then, the development of continuum kinematics, stress and equilibrium equations is followed by considering hyperelasticity as the material constitutive model. After a discussion Saint Venant model arises as the suitable for the wing model. Once the set of equations is obtained a linearization phase occurs before the final discretization. The solution is achieved by using the Newton-Raphson scheme.

The computer implementation is based on test-driven development and it grows concurrently with the results. Cases with geometries of increasingly complexity are solved until reaching the wing case.



# NOMENCLATURE

$F$	Deformation gradient tensor
$C$	Right-Cauchy strain tensor
$b$	Left-Cauchy strain tensor
$E$	Green-Lagrange strain tensor
$e$	Eulerian strain tensor
$U$	Material stretch tensor
$V$	Spatial stretch tensor
$\tau$	Kirchhoff stress tensor
$P$	First Piola-Kirchhoff stress tensor
$S$	Second Piola-Kirchhoff stress tensor
$\sigma$	Cauchy stress tensor
$\dot{E}$	Strain rate tensor
$d$	Deformation tensor
$\dot{F}$	Time derivative of $F$
$R$	Residual force
$T$	Internal force
$F$	External force
$K$	Tangent matrix
$K_\sigma$	Geometric tangent matrix component
$K_c$	Constitutive tangent matrix component
$K_p$	Pressure tangent matrix component
$\Psi$	Stored strain energy function
$\dot{E}$	Green-Lagrange strain rate
$\varepsilon$	Linear strain
$C$	Lagrangian or material elasticity tensor
$c$	Eulerian or spatial elasticity tensor
$\mathcal{U}$	Strain energy density
$A$	Cross section Area
$E$	Young's modulus
$q$	Distributed load
$P$	Point load
$U$	Internal energy
$W$	External work potential

$u$	General displacement
$u^*$	Exact displacement solution
$u^e$	Finite element trial function
$a$	Node index
$e$	Element index
$\phi_i^e$	Basis function of node $i$ and element $e$
$m$	Number of elements
$n$	Number of nodes per element
$N_a^e$	Shape Function of element $e$ in node $a$
$\mathbf{K}^e$	Element stiffness matrix
$\mathbf{F}^e$	Consistent nodal
$\mathbf{E}$	Plane stress elastic moduli matrix
$\mathbf{N}$	Shape function matrix
$\mathbf{D}$	Symmetric-gradient operator
$\mathbf{B}$	Strain-displacement matrix
$\lambda$	First lame parameter
$\mu$	Second Lamé parameter or tangent module
$\hat{\mathbf{u}}$	Prescribed displacements
$\hat{\mathbf{t}}$	Prescribed tractions
$\hat{\mathbf{p}}$	Prescribed forces
$\mathcal{I}$	Fourth-order identity tensor

## **ACKNOWLEDGMENTS**

First and foremost, my sincere gratitude to my advisor Professor Àlex Ferrer Ferré for the opportunity of conducting my master thesis under his guidance. I appreciate his patience to teach no matter the difficulty of the concept or the time needed, and his contagious motivation which made me overcome any type of problem, especially during my stay of five months in Sweden. Thank you for provided me with the correct direction.

Besides my advisor, I would also like to thank my friend and one of the *FEM-MAT-OO* contributors Raül. Thank you for the months you had to endure living with me during our stay in Sweden, for listening and sharing views.

Last but not least, I thank the invaluable support of my family. To my parents Manuel and Lina, who always encouraged me to pursuit my goals, for the education and guidance they provided me. To my grandparent Cristóbal who taught me from a very young age how to solve problems with ingenious. To my sisters Claudia, Carlota and Carolina for support me. Finally, to my partner Irene for the love and support.

# CONTENTS

<b>Nomenclature</b> . . . . .	<b>v</b>
<b>Acknowledgments</b> . . . . .	<b>VII</b>
<b>1. Introduction</b> . . . . .	<b>1</b>
<b>1.1. Motivation of the Project</b> . . . . .	<b>1</b>
<b>1.2. Objectives</b> . . . . .	<b>3</b>
<b>1.3. Scope</b> . . . . .	<b>3</b>
<b>2. Nonlinear Finite Element Method</b> . . . . .	<b>5</b>
<b>2.1. Brief introduction to FEM</b> . . . . .	<b>5</b>
<b>2.2. Nonlinear Computational Mechanics</b> . . . . .	<b>8</b>
2.2.1. Geometric Nonlinearity . . . . .	8
2.2.2. Nonlinear Strain Measures . . . . .	9
<b>2.3. Kinematics</b> . . . . .	<b>16</b>
2.3.1. The Motion . . . . .	16
2.3.2. Material and Spatial Descriptions . . . . .	17
2.3.3. Deformation Gradient . . . . .	18
2.3.4. Deformation and Strain . . . . .	21
2.3.5. Polar Decomposition . . . . .	24
2.3.6. Volume & Area Change . . . . .	25
2.3.7. Linearized Kinematics . . . . .	27
2.3.8. Velocity and Material Time Derivatives . . . . .	30
2.3.9. Rate of Deformation . . . . .	31
2.3.10. Objectivity . . . . .	33
<b>2.4. Stress &amp; Equilibrium</b> . . . . .	<b>35</b>
2.4.1. Cauchy Stress Tensor . . . . .	35
2.4.2. Equilibrium . . . . .	36
2.4.3. Principle of Virtual Work . . . . .	38
2.4.4. Work Conjugacy and Alternative Stress Representations . . . . .	39
2.4.5. Physical Interpretation of Stresses . . . . .	42
<b>2.5. Constitutive Model</b> . . . . .	<b>44</b>

2.5.1. Hyperelasticity . . . . .	45
2.5.2. Elasticity Tensor . . . . .	46
2.5.3. Isotropic Hyperelasticity . . . . .	47
2.5.4. Constitutive Material Models . . . . .	48
<b>2.6. Linearized Equilibrium Equations . . . . .</b>	<b>50</b>
2.6.1. Linearization and Newton-Raphson Process . . . . .	50
2.6.2. Lagrangian Linearized Internal Virtual Work . . . . .	51
2.6.3. Eulerian Linearized Internal Virtual Work . . . . .	52
2.6.4. Linearized External Virtual Work . . . . .	53
<b>2.7. Discretization &amp; Solution . . . . .</b>	<b>55</b>
2.7.1. Discretized Kinematics . . . . .	56
2.7.2. Discretized Equilibrium Equations . . . . .	58
2.7.3. Discretization of the Linearized Equilibrium Equations . . . . .	60
2.7.4. Numerical Methods . . . . .	63
<b>3. Computer Implementation . . . . .</b>	<b>69</b>
<b>3.1. Object Oriented Programming . . . . .</b>	<b>69</b>
3.1.1. Object-Oriented Elements . . . . .	69
3.1.2. Object-Oriented Principles . . . . .	70
<b>3.2. FEM-MAT-OO . . . . .</b>	<b>72</b>
3.2.1. Code Instructions . . . . .	72
3.2.2. Class organization . . . . .	73
3.2.3. Test-driven Development . . . . .	74
3.2.4. Hyperelasticity Core . . . . .	75
3.2.5. Source Control . . . . .	76
<b>4. Results . . . . .</b>	<b>77</b>
<b>4.1. Test Problem . . . . .</b>	<b>77</b>
<b>4.2. Preliminary Cases . . . . .</b>	<b>78</b>
4.2.1. 2D Cases . . . . .	78
4.2.2. 3D Cases . . . . .	81
<b>4.3. Wing Loading Test . . . . .</b>	<b>86</b>
4.3.1. Regulations & Procedure . . . . .	86
4.3.2. Structural Design . . . . .	87
4.3.3. Meshing . . . . .	87
4.3.4. Boundary Conditions . . . . .	88
4.3.5. Simulation Results . . . . .	89

<b>Conclusions</b>	<b>93</b>
<b>Bibliography</b>	<b>95</b>
<b>A. Mathematical Formulation of Finite Elements</b>	<b>99</b>
<b>A.1. Bar Element</b>	<b>99</b>
A.1.1. Variational Formulation	99
A.1.2. The Galerkin Method	105
<b>A.2. Plane Stress Problem</b>	<b>109</b>
A.2.1. Plate in Plane Stress	109
A.2.2. Governing Equations	111
A.2.3. Boundary Conditions	112
A.2.4. Finite Element Equations	112
<b>A.3. Three-Node Plane Stress Triangles</b>	<b>114</b>
A.3.1. Triangular Coordinates	115
A.3.2. Linear Interpolation	116
A.3.3. Coordinate Transformations	116
A.3.4. Partial Derivatives	116
A.3.5. Turner Triangle	117
A.3.6. Isoparametric Triangles	119
<b>A.4. Isoparametric Quadrilaterals</b>	<b>120</b>
A.4.1. Isoparametric Representation	120
A.4.2. Quadrilateral Coordinates	122
A.4.3. Bilinear Quadrilateral	122
A.4.4. Partial Derivative Computation	123
A.4.5. Partial Derivatives	123
A.4.6. Numerical Integration	125

# LIST OF FIGURES

1.1.	A static load test of the Lockheed Vega wing in 1929 . . . . .	1
1.2.	Ultimate-load wing up-bending test on the 787 Dreamliner static test unit. . . . .	2
2.1.	Fixed-free bar with a punctual load applied in the right-end. The mesh is formed by 7 two-noded line elements. Nodes have the function of defining both the element geometry and the DOF. . . . .	5
2.2.	Displacement function $u$ is approximated by a trial function $u^e$ . . . . .	6
2.3.	Approximation of a 2D vector by a 1D vector in the vector space $V$ . . . . .	7
2.4.	Cantilever; (a) Free body diagram; (b) Linear and nonlinear solutions. . . . .	8
2.5.	One-Dimensional strain. . . . .	9
2.6.	Single incompressible truss member. . . . .	11
2.7.	Truss example: load deflection behaviour. . . . .	12
2.8.	Truss member example for different initial angles. . . . .	12
2.9.	90° rotation of a two-dimensional body. . . . .	14
2.10.	General deformation of a two-dimensional body. . . . .	15
2.11.	General motion of a deformable body. . . . .	17
2.12.	General motion in the neighborhood of a particle. . . . .	18
2.13.	Simple deformations of a two-dimensional body. . . . .	20
2.14.	Change in length. . . . .	22
2.15.	Volume change. . . . .	26
2.16.	Area change. . . . .	27
2.17.	Linearized kinematics. . . . .	28
2.18.	Lie derivative diagram. . . . .	33
2.19.	Superimposed rigid body motion. . . . .	33
2.20.	Elemental tetrahedron. . . . .	35
2.21.	Interpretation of stress tensors. . . . .	41
2.22.	Cauchy and second Piola-Kirchhoff stress for beam. . . . .	43
2.23.	Cauchy and second Piola-Kirchhoff stress distributions. . . . .	44
2.24.	Force versus displacement for Saint Venant-Kirchhoff and Neo-Hookean models. . . . .	48
2.25.	Uniform surface pressure. . . . .	54
2.26.	Newton-Raphson iteration. . . . .	65
2.27.	Response diagrams: (a) typical load-deflection diagram showing equilibrium path; (b) diagram distinguishing primary from secondary equilibrium path. . . . .	66
2.28.	(a) Snap-back; (b) Spherical arc-length method. . . . .	67
3.1.	Definition of a class as the highest level of sophistication. . . . .	69
3.2.	<i>Element</i> class as a template to make (instantiate) objects. . . . .	70
3.3.	An example of inheritance; the <i>Element</i> class. . . . .	72
3.4.	<i>FEM-MAT-OO</i> UML diagram showing the main classes related to non-linear FEM. . . . .	73
3.5.	Diagram showing the dependencies of the most important quantities related to hyperelastic analysis. . . . .	75

3.6.	<i>FEM-MAT-OO</i> network graph. . . . .	76
4.1.	Simple two-dimensional example. . . . .	77
4.2.	Load-displacement curve of node 4. . . . .	78
4.3.	Deformation of body. . . . .	78
4.4.	Force versus displacement of node 48. . . . .	79
4.5.	Load-displacement curve of node 48. . . . .	79
4.6.	Distribution of $\sigma_x$ for different increments. . . . .	80
4.7.	Mesh diagram and response diagram monitoring node 6. . . . .	81
4.8.	Deformation process of a quadrilateral subjected to a normal force. . . . .	81
4.9.	Deformation process of a quadrilateral subjected to an axial force. . . . .	81
4.10.	Mesh and boundary conditions of Hexahedron I case. . . . .	82
4.11.	Hexaedron deformation process. . . . .	82
4.12.	Response diagrams of hexaedron cases II and III. . . . .	83
4.13.	Deformation process of a hexaedron subjected to normal forces. . . . .	84
4.14.	Deformation process of a hexaedron subjected to normal forces. . . . .	84
4.15.	Boundary conditions and response diagram of slender case. . . . .	85
4.16.	$\sigma_{xy}$ stress-component distribution for the slender beam case. . . . .	85
4.17.	$\sigma_{xy}$ stress-component distribution showing the lower part of the beam. . . . .	86
4.18.	Boeing 737-800 plan view schematic. . . . .	87
4.19.	Wing design and meshing process. . . . .	88
4.20.	Wing boundary conditions. . . . .	88
4.21.	Root airfoil. Each airfoil is composed by 45 nodes. Maximum thickness is 14 cm. . . . .	89
4.22.	Superimposed deflection of the wing. . . . .	89
4.23.	Response diagram of the wing. . . . .	89
4.24.	Deformation of the wing. . . . .	90
4.25.	Final $e_y$ strain-component distribution at the extrados. . . . .	90
4.26.	Evolution $\sigma_y$ stress-component distribution at the extrados. . . . .	91
4.27.	Final $\sigma_y$ stress-component distribution at the intrados. . . . .	91
A.1.	Concept of admissible variation of the axial displacement function $u(x)$ . .	101
A.2.	Fem discretization of a fixed-free bar member into four two-node elements. .	101
A.3.	A two-node, TPE-based bar element. . . . .	102
A.4.	A plate structure in plane stress. . . . .	109
A.5.	Displacement and force (stress, traction) BC for the plane stress problem. .	112
A.6.	The TPE-based Weak Form of the plane stress equations. . . . .	113
A.7.	Three-node, linear-displacement plane stress triangular element. . . . .	115
A.8.	Triangular coordinates $\zeta_1, \zeta_2, \zeta_3$ . . . . .	115
A.9.	Displacement interpolation over triangle. . . . .	118
A.10.	The 3-node linear triangle. . . . .	120
A.11.	The 6-node quadratic triangle. . . . .	120
A.12.	Isoparametric representation of arbitrary 2D elements. . . . .	121
A.13.	Quadrilateral coordinates. . . . .	122
A.14.	First five 1D Gauss rules pictured over the segment. . . . .	125
A.15.	First four 2D Gauss rules $P = 1, 2, 3, 4$ pictured over a straight-sided quadrilateral region. . . . .	126
A.16.	Geometric interpretation of the Jacobian determinant role. . . . .	127

# LIST OF TABLES

2.1.	Significance of $\mathbf{u}$ and $\mathbf{f}$ depending on the physical problem. . . . .	5
2.2.	Material and spatial notation. . . . .	16
2.3.	Deformation gradient examples. . . . .	21
2.4.	Characteristics of a general deformable body in 3D. . . . .	37
2.5.	Equilibrium equations. . . . .	43
2.6.	Résumé of all conjugate stress-strain tensor pairs. . . . .	44
3.1.	Main properties of Geometry class. . . . .	74
3.2.	Element_Hyperelastic child implementation. . . . .	75
4.1.	Summarized parameters. . . . .	77
4.2.	Summarized parameters of the strip-hole case. . . . .	79
4.3.	Summarized parameters of the quadrilateral case. . . . .	80
4.4.	Summarized parameters of the hexaedron case. . . . .	83
4.5.	Summarized parameters of the hexaedron II case. . . . .	83
4.6.	Summarized parameters of the hexaedron III case. . . . .	84
4.7.	Summarized parameters of the hexaedron case. . . . .	85
4.8.	Comparison between model and Boeing 737-800 data. . . . .	87
4.9.	Summarized parameters of the wing case. . . . .	88
A.1.	One-dimensional Gauss rules with 1 through 3 sample points. . . . .	125

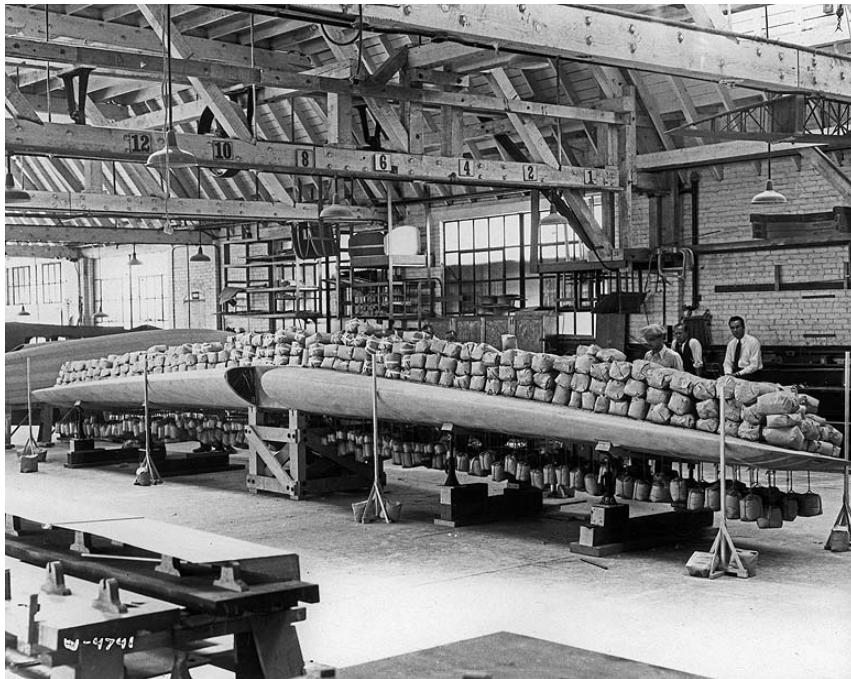


# CHAPTER 1. INTRODUCTION

## 1.1. Motivation of the Project

### The Wing Static Test

Ultimate wing load testing<sup>1</sup> is one of the main tests an aircraft must rigorously pass during the wing design process. This standard procedure is performed by the manufacturers for any new airplane design and it is an important achievement towards the certification of a civil aircraft. Whether the airplane is large or small, this procedure has been done since the beginning of aviation.



**Figure 1.1:** A static load test of the Lockheed Vega wing in 1929 [1].

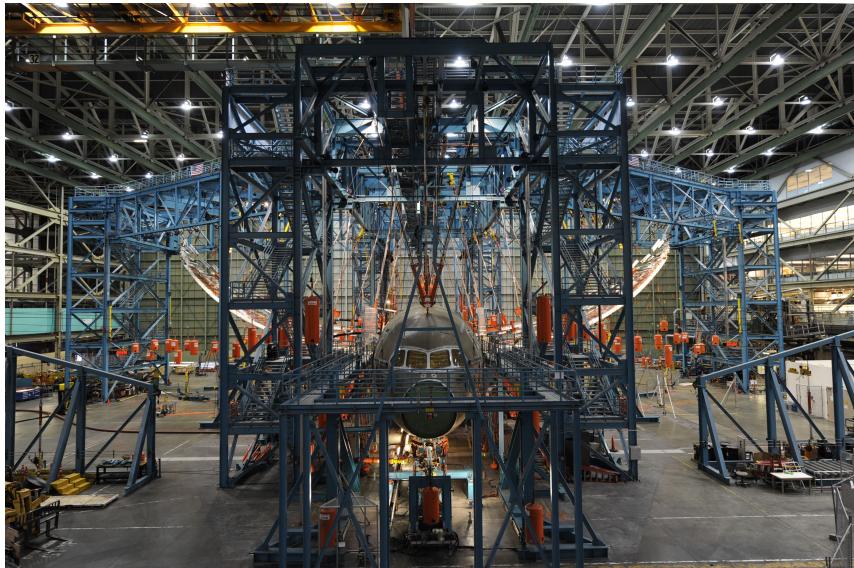
In the early days of aviation simple sandbags were placed on the wing of an aircraft to represent the maximum forces needed to be tested. This procedure evolved towards hydraulic jacks where the measures were read from rudimentary scales. Nowadays these jacks have been replaced by actuators with load cells to control the applied load. In addition, the reads are made by using strain gauges and lasers to measure the wing deflection.

### Nonlinear Simulations

Due to its need for solving complex design problems, the aerospace industry emerged as one of the main users of nonlinear simulations. It is primarily during the phases related with

---

<sup>1</sup>Various terms are employed when referring to wing load testing. Wing-up bending test or wing static test are also typically used in the aviation jargon.



**Figure 1.2:** Ultimate-load wing up-bending test on the 787 Dreamliner static test unit. The wings were flexed upward by approximately 25 feet (7.6 meters), having a wing span of 60 meters [2].

structure design processes where nonlinear simulations take place. At present, simulations not only secure structural tests, but also complement them during all its lifecycle in three different ways.

- **Prior** to test. To avoid premature failures during test.
- **During** test. To explain and rectify issues encountered during the test execution.
- **Post** test. To explain test failures and analyse modifications.

All of the abovementioned phases help manufacturers to save time and reduce design cycles, which in turn means reducing costs.

The design process could be separated into two parts. In the first part, a topology, sizing and shape optimisation process is performed in an attempt to achieve lighter and more efficient structure designs. In the second part, these designs are subjected different tests, such as bending or vibration, to verify they meet the stress and stiffness requirements.

Over the coming years, simulations will not replace aircraft structural tests owing to the fact that computers are not yet powerful enough to predict the most dramatic material property, fracture toughness [3]. A huge amount of information about the material's atomic structure over large distances is needed to forecast fracture. However, simulation will play more and more a major role in the aircraft process, from the development of new concepts to the support of the in-service fleet.

In this context of nonlinear simulations the Finite Element Method (FEM) arises as the most powerful tools available in the solution of problems related to structural analysis.

## 1.2. Objectives

The aim of this study is to develop a collaborative software, called *FEM-MAT-OO*<sup>2</sup>, based on the object-oriented programming (OPP) paradigm. This software is expected to solve a large variety of problems inside the FEM framework.

Inside *FEMATOOL*, the purpose of this work is to implement nonlinear structural analysis. The formulation of FEM by variational principle is firstly applied to linear continuum mechanics, then implemented in MATLAB® and finally validated by several tests.

Once the familiarization period is established, the finite-strain structural problem is variationally formulated. Focus is posed in nonlinearities, geometry, motion description and redefinition of stress and strain tensors. Additionally, discretization of the variational problem and nonlinear algorithms are studied and developed.

To accurately simulate the deformation of a wing a proper constitutive model needs to be formulated. The most common simplifications, such as compressibility, isotropy and elastic behaviour of the material, are assumed. Different models are studied and implemented in order to properly fit with the real aircraft structure.

In order to achieve this objective, well-defined steps with marked milestones are established in order not to divert from the original target. These milestones are summarized as follows:

1. Developement of FEM formulation for solving linear elasticity.
2. Developement of nonlinear FEM formulation to solve geometries of increasing complexity.
3. Identification of the constitutive model and material parameters which suits the wing wing behaviour.

## 1.3. Scope

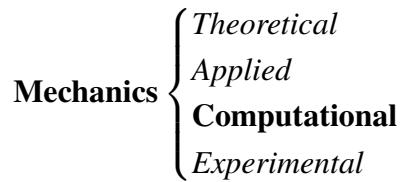
Aircraft nonlinear simulations are performed in different Physics areas such as thermodynamics or electromagnetism. Nevertheless, it is mechanics, and more specifically continuum mechanics, the branch which embraces the main topics of the project. This scientific discipline studies the effect of forces and energy on physical bodies<sup>3</sup>, by having as a prime application the design and construction of material systems, such as vehicles or structures.

In order to contextualize how the nonlinear continuum analysis has been implemented in the FEM environment, a top-down classification is done. Mechanics embraces the following four areas:

---

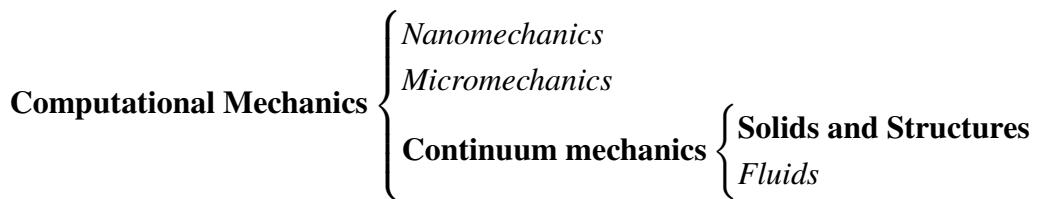
<sup>2</sup>*FEMATOOL* stands for Finite Element method MATLAB Object Oriented.

<sup>3</sup>In continuum mechanics the term *bodies* applied to materials modeled as a continuous mass.



While *Theoretical mechanics* is related to fundamental laws and principles, *Applied mechanics* transfers this theoretical knowledge to scientific and engineering applications, specially as regards the construction of mathematical model of physical phenomena. *Computational mechanics* solves specific problems by model-based simulation through numerical methods implemented on computers [4]. *Experimental mechanics* deals with the observation of physical phenomena and experiments.

Computational mechanics can be classified depending on the physical scale under study:



While *nanomechanics* deals with phenomena at molecular and atomic level, *micromechanics* studies the behaviour of composites in heterogeneous materials. The former is mainly used in particle physics and chemistry; the latter, though, has its main technological application in the design and fabrication of materials and microdevices . *Continuum mechanics* studies bodies at macroscopic level in which the microstructure is homogenized by phenomenological averaging [4].

Although there are many different methods to discretize and solve the PDEs raised in continuum mechanics, in the case of nonlinear continuum mechanics, the use of FEM is dominant.

# CHAPTER 2. NONLINEAR FINITE ELEMENT METHOD

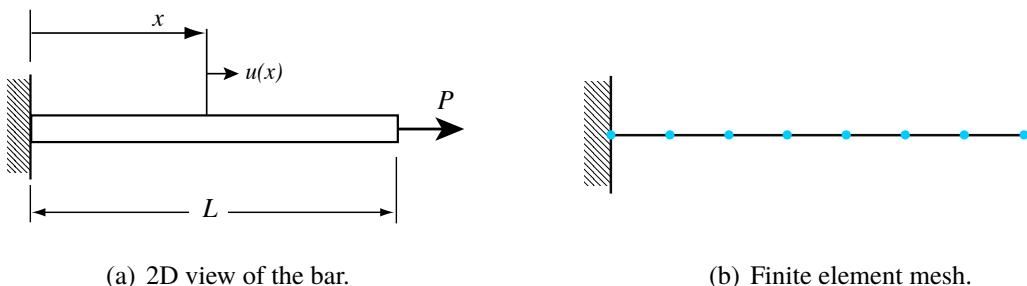
## 2.1. Brief introduction to FEM

The vast majority of physical problems can be mathematically modelled by a partial differential equation (PDE). Often, these equations either do not have analytical solution, or their solution is so complicated that finding it is impracticable. In this context a numerical approximation to the unknown solution is done. Hence, the finite element method is nothing but a tool for solving PDEs.

<i>Application Problem</i>	<i>Vector <math>u</math> represents</i>	<i>Vector <math>f</math> represents</i>
<b>Solids and structures</b>	<b>Displacement</b>	<b>Mechanical force</b>
Heat conduction	Temperature	Heat flux
Potential flows	Pressure	Particle velocity
General flows	Velocity	Fluxes
Electrostatics	Electric potential	Charge density
Magnetostatics	Magnetic potential	Magnetic intensity

**Table 2.1:** Significance of  $u$  and  $f$  depending on the physical problem [5].

FEM is based in three well-defined steps: idealization, discretization and solution. The idealization is basically the mathematical modeling expressed as a PDE. After this simplification step, a model with an infinite number of degrees of freedom (DOF) is obtained. The action of transforming a continuous model to a discrete one is called discretization. The DOFs are the number of parameters of the system that may vary independently. For instance, a point in a two-dimensional plane has two DOFs for translation in both  $x$  and  $y$  coordinates<sup>1</sup>.

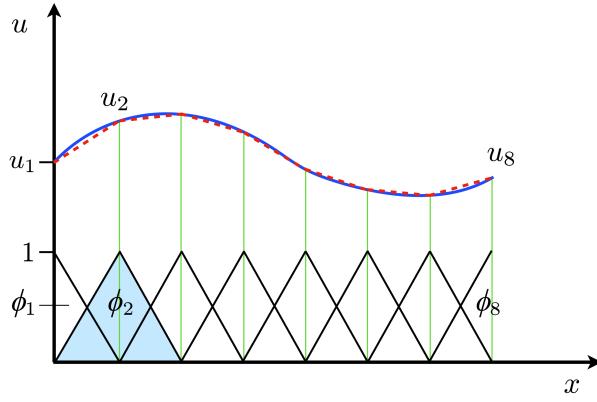


**Figure 2.1:** Fixed-free bar with a punctual load applied in the right-end. The mesh is formed by 7 two-noded line elements. Nodes have the function of defining both the element geometry and the DOF.

There are different numerical methods for solving PDE, such as the finite difference method (FDM) or the finite volume method (FVM). One of the most noticeable differences is that

<sup>1</sup>Degrees of freedom are also called generalized coordinates.

while in FDM and FVM a numerical approximation of the governing equation is proposed, in FEM it is the solution which is approximate.



**Figure 2.2:** The function  $u$  (solid blue line) is approximated with  $u^e$  (dashed red line), which is a linear combination of linear basis functions  $\phi_i$  (solid black lines). The coefficients, which are the nodal displacements are denoted by  $u_1$  through  $u_8$ .

Considering the example of Figure 2.1, the axial displacement  $u(x)$  of the bar can be approximated by a function  $u^e(x)$  using linear combinations of basis functions according to the following expressions

$$u \approx u^e; \quad u^e = \sum_{i=1}^m \phi_i u_i, \quad (2.1)$$

where  $\phi_i$  denotes the basis functions and  $u_i$  represents the coefficients of the functions that approximate  $u$  with  $u^e$ . The coefficients  $u_1, \dots, u_m$  are to be determined, being  $m$  the number of elements. Figure 2.2 depicts a graphical way of understanding the relationship between the basis functions and the nodal displacements. Note that in this case a regular discretization of the space is performed. Nevertheless, another benefit of FEM is the possibility of employ smaller or larger elements when needed. To compute the coefficients or nodal displacements, it will be necessary a first step to construct  $m$  equations, and then a second step for determining these unknown coefficients. Thus, the equation to be solve is

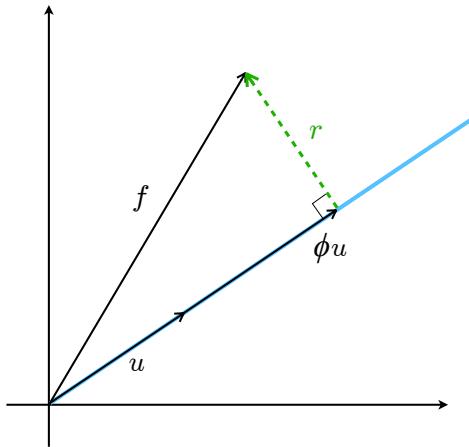
$$K u^e = f, \quad (2.2)$$

where  $f$  is the given force vector of dimension 1 to  $m$ , and  $K$  is the stiffness matrix (even in non-structural applications) which contains the coefficients of  $u_i$  in each equation  $j$  within its components  $K_{ji}$ .

Although multiple approximation methods can be used, the Galerkin method is the one used in this work. As an approximation method, the goal is to reduce the difference between the left and right-hand side of Equation 2.2. Rearranging the terms in the same side as follows

$$r = Ku^e - f = 0, \quad (2.3)$$

where  $r$  is called the residual force, which for the purpose can be seen as an *error*. Galerkin method is based on finding  $u^e$  such that the residual  $r$  is orthogonal to the space where  $u^e$  is seek. Actually, we are looking for the projection of vector  $f$  since any other vector will lead to a larger distance between  $f$  and  $u^e$  (see Figure 2.3). Equation 2.4 represents the condition which must be satisfied to ensure the orthogonality of the residual.



**Figure 2.3:** Approximation of a two-dimensional vector by a one-dimensional vector in the vector space  $V$ . As matrix  $K$  represents a linear map between displacement vector and force vector is not depicted in the figure due to simplification issues.

$$r \cdot v = 0 \quad \forall v \in V. \quad (2.4)$$

Searching the *residual* to be orthogonal to the space  $V$  is the reason why Galerkin belongs to weighted residual techniques.

The vector space  $V$ , spanned<sup>2</sup> by the vector  $\phi$ , is defined mathematically by  $V = \text{span}\{\phi\}$ . Although Galerkin is founded in approximating functions, both functions and vectors are equivalent and valid for explaining the method. Hence, minimizing the error is equivalent to demanding  $r$  to be orthogonal to any test vector  $v \in V$ , called test function in Appendix A by analogy. Truly, the Galerkin method only applies when test  $v$  function and trial function  $u^e$  bases coalesce.

The standard approach to deriving the Galerkin scheme implies to multiply both sides of Equation 2.3, and to integrate over the domain giving

$$\int_0^L r(x) v(x) dx = 0. \quad (2.5)$$

To achieve the condition  $r = 0$  in all the points along the bar can be too demanding. The integration allow us to satisfy the condition only in an average sense.

<sup>2</sup>The span the set of all possible vectors you can reach with a linear combination of a given set of vectors depending on the dimension; two vectors in  $\mathbb{R}^2$ .

The complete linear FEM formulation is derived in Appendix A. First a variational formulation based on the physical principle of minimizing energy is introduced. Then, the Galerkin approximation method is used to find the same solution.

## 2.2. Nonlinear Computational Mechanics

There are two main sources of nonlinearity in the analysis of solid continua, namely material and geometric nonlinearity. The former occur when the constitutive relation between stress and strain is nonlinear, while the latter takes place when there is a significant change in the geometry.

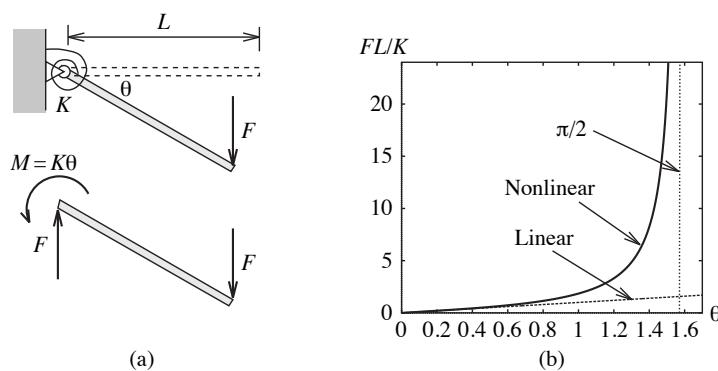
The classical development of finite elements is under the framework of linear elasticity. It is obvious the success of the assumption of linearity in engineering analysis, however, it is equally self-evident that many situations demand consideration of nonlinear behaviour. For example, ultimate load analysis of structures involves material nonlinearity and perhaps geometric nonlinearity, and any metal-forming analysis such as forging or crash-worthiness must include both aspects of nonlinearity. In fact, the mechanical behaviour of the human body itself involves both types of nonlinearity [6].

Both nonlinear and linear continuum mechanics deal with the same topics such as kinematics, stress and equilibrium, and constitutive behaviour. Nevertheless, in the linear case an assumption is made that the deformation is sufficiently small to ignore the effect of changes in the geometrical configuration of the solid.

Owing to the nature of the work the geometric nonlinearity will be the unique source nonlinearity. Therefore, it means that the material properties do not vary depending on the deformation state.

### 2.2.1. Geometric Nonlinearity

To introduce some important concepts that will be used in future sections, a simple geometrically nonlinear example is presented below.



**Figure 2.4:** Cantilever; (a) Free body diagram; (b) Linear and nonlinear solutions.

This cantilever example illustrates a simplified model of the wing static test. Here, a weightless rigid bar-linear elastic torsion spring model is used (Figure 2.4). Taking

moments about the hinge gives the equilibrium equation as

$$FL \cos \theta = M. \quad (2.6)$$

Torsion springs obey the rotational form of Hooke's Law as

$$M = K \theta. \quad (2.7)$$

Here  $K$  is the torsional stiffness of the spring. Substituting the latter equation into 2.6 we obtain the following nonlinear relationship.

$$\frac{FL}{K} = \frac{\theta}{\cos \theta}. \quad (2.8)$$

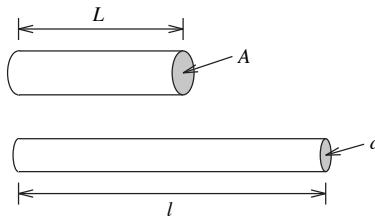
If the angle  $\theta \rightarrow 0$ , then  $\cos \theta \rightarrow 1$ , and the linear equation is stated as

$$F = \frac{K}{L} \theta. \quad (2.9)$$

The exact nonlinear equilibrium path is shown in Figure 2.4(b), where no physical solution makes sense for  $\theta > \pi/2$ . In this example is seen that only by considering finite displacements of the structure a nonlinear solution has been achieved.

### 2.2.2. Nonlinear Strain Measures

In the previous section, the beam remained rigid during the deformation. In general, continuous bodies will exhibit large strains when undergoing a geometrically nonlinear deformation process. These strains, as well as the difficulties involved in the definition, will be discussed in this section.



**Figure 2.5:** One-Dimensional strain [6].

#### 2.2.2.1. One-Dimensional Strain Measures

The most common definition of strain is the so-called *engineering* strain  $\varepsilon_E$ . Considering the truss member, depicted in Figure 2.5, of initial length  $L$  and area  $A$  that is stretched to a final length  $l$  and area  $a$ , the engineering strain is defined as

$$\varepsilon_E = \frac{l - L}{L}. \quad (2.10)$$

Here, the change in length  $\Delta l = l - L$  is divided by the initial length. However, different strain measures can be used. For instance, the change in length could be divided by the final length  $l$ . For  $l \approx L$  the small strain quantity  $\varepsilon = \Delta l/l$  is recovered.

Alternative large strain measures can be obtained by taking into account the whole deformation process, i.e. adding up all the small strain increments that take place from its original to its final length. The integration process leads to the definition of the *logarithmic* or *natural* strain  $\varepsilon_L$  as

$$\varepsilon_L = \int_L^l \frac{dl}{l} = \ln \frac{l}{L}. \quad (2.11)$$

Although the above strain definitions can be extrapolated to the deformation of a three-dimensional continuum body, this generalization process is complex and computationally costly [6]. In this context, two strain cases much more readily generalized to continuum cases appear. These strains are called *Green* strain  $\varepsilon_G$  and *Almansi* strain  $\varepsilon_A$  defined as

$$\varepsilon_G = \frac{l^2 - L^2}{2L^2}; \quad (2.12a)$$

$$\varepsilon_A = \frac{l^2 - L^2}{2l^2}. \quad (2.12b)$$

For the case where  $l \approx L$  it can be demonstrated that both Green and Almansi strain quantities converge towards to the small strain definition  $\Delta l/l$  by using a simple Taylor series. For example, the Green strain case gives

$$\varepsilon_G \approx \frac{(L + \Delta l)^2 - L^2}{2L^2} = \frac{L^2 + \Delta L^2 + 2L\Delta l - L^2}{2L^2} = \frac{\Delta l}{L} + \frac{1}{2} \left( \frac{\Delta l}{L} \right)^2. \quad (2.13)$$

#### 2.2.2.2. Nonlinear Truss Example

This example will serve us to introduce the nonlinear equilibrium equation, which is used extensively during the work. Considering the truss member shown in Figure with initial and loaded lengths, cross-sectional areas and volumes:  $L, A, V$  and  $l, a, v$  respectively. For simplicity, the material is assumed incompressible and hence  $V = v$ .

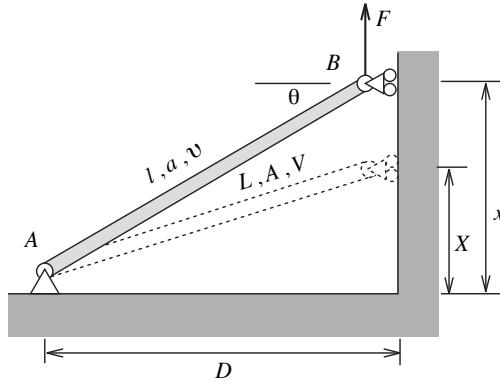
For the moment, two constitutive equations are chosen on Green's and logarithmic definition of strain. Therefore, the Cauchy stress  $\sigma$  is either

$$\sigma = E \frac{l^2 - L^2}{2L^2}; \quad \sigma = E \ln \frac{l}{L}, \quad (2.14)$$

where  $E$  is a constitutive constant<sup>3</sup> that, as an assumption, has been chosen to be the same irrespective of the strain measure chosen. In fact, this is only acceptable for small strains. The assumption will be discussed in Section 2.4.

---

<sup>3</sup>In this case  $E$  can be seen as a Young's modulus like constitutive constant.



**Figure 2.6:** Single incompressible truss member [6].

The equation for vertical equilibrium at joint *B* is

$$R(x) = T(x) - F, \quad (2.15)$$

where  $T(x)$  is the vertical component of the internal force in the truss member and it is computed as

$$T = \sigma a \sin \theta; \quad \sin \theta = \frac{x}{l}, \quad (2.16)$$

and  $R(x)$  is the *residual* or *out-of-balance* force, and a solution for  $x$  is achieved when  $R(x) = 0$ . In terms of alternative strain measures, Equation 2.16 is expressed as

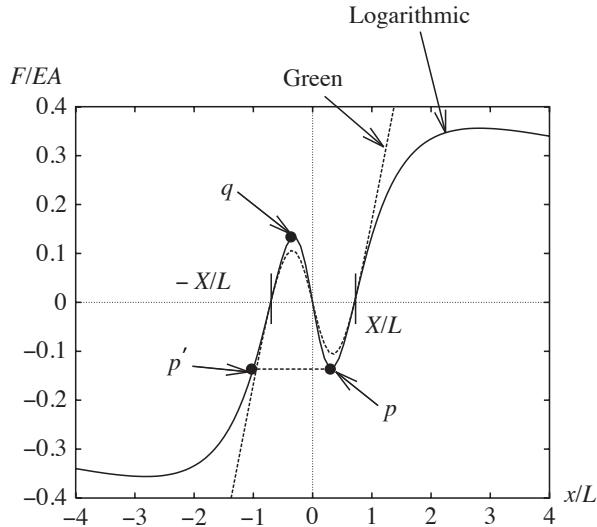
$$T = \frac{E v x}{l^2} \left( \frac{l^2 - L^2}{2L^2} \right) \quad \text{or} \quad T = \frac{E v x}{l^2} \ln \frac{l}{L}. \quad (2.17)$$

Note that the nonlinearity of  $T$  in previous equations arises from the fact that  $l$  is a function of  $x$  as  $l^2 = D^2 + x^2$ , being  $T$  highly nonlinear in  $x$ . Typically, given a value of the external load  $F$ , the position value is obtained by using the Newton-Raphson method. However, for this one-degree-of-freedom case it is easier to impose a value for  $x$ . For the common case of  $\theta_0 = X/L = 45^\circ$  the results are shown in Figure 2.7.

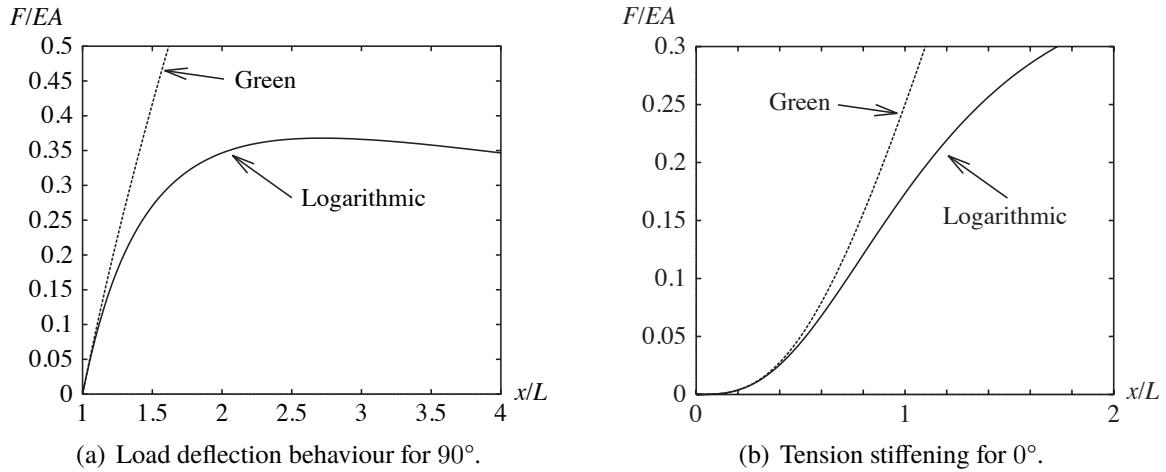
It is obvious that different constitutive relations will give different results, and thus the selection of strain measure has to be taken carefully. However, it is seen that both Green and logarithmic results, in the region near 1 where the strains are to be small, have similar equilibrium paths. In Figure 2.7, the local maximum and minimum forces  $F$  occur at points *p* and *q*. These points are called *limit points*. In the case the truss was compressed to point *p* it would experience a violent movement from *p* to point *p'* as an attempt is made to increase the compressive load in the truss beyond the limit point [6]. This backward movement in the load deflection behaviour is also known as *snap-through behaviour*.

The load deflection behaviour of the truss member from a vertical initial position is shown in Figure 2.8(a). It is seen that  $E$  can not be used to represent the same material characterized using different strain measures. Alternatively, Figure 2.8(b) shows the stiffening behaviour of the truss member from an horizontal initial position.

As it is said before, the solution for  $x$  is achieved when  $R(x) = 0$ . The stiffness of the structure is generally represented by  $K = dR/dx$ . If the load  $F$  is constant, the stiffness is



**Figure 2.7:** Truss example: load deflection behaviour for an initial angle of 45° [6].



**Figure 2.8:** Truss member example for different initial angles [6].

the change of the vertical component  $T$  of the internal force. By using Equations 2.15 and 2.16, together with the incompressibility condition  $a = V/l$ , the stiffness is computed by using the chain rule as

$$\begin{aligned}
 K &= \frac{dT}{dx} \\
 &= \frac{d}{dx} \left( \frac{\sigma V x}{l^2} \right) = \frac{d}{dx} \left( \frac{\sigma a x}{l} \right); \\
 &= \left[ \frac{a x d\sigma}{l} - \frac{2l \sigma V x}{l^4} \right] \frac{dl}{dx} + \frac{\sigma a}{l} \\
 &= \left[ \frac{a x d\sigma}{l} - \frac{2\sigma a x}{l^2} \right] \frac{dl}{dx} + \frac{\sigma a}{l}.
 \end{aligned} \tag{2.18}$$

The remaining step is to find  $d\sigma/dl$  for each strain definition. Thus, for Green's and logarithmic strain, respectively, we compute

$$\left(\frac{d\sigma}{dl}\right)_G = \frac{El}{L^2} \quad \left(\frac{d\sigma}{dl}\right)_L = \frac{E}{l}. \quad (2.19)$$

Hence the stiffnesses are

$$K_G = \frac{A}{L} \left( E - 2\sigma \frac{L^2}{l^2} \right) \frac{x^2}{l^2} + \frac{\sigma a}{l} \quad (2.20a)$$

$$K_L = \frac{a}{l} \left( E - 2\sigma \frac{x^2}{l^2} \right) + \frac{\sigma a}{l}. \quad (2.20b)$$

It is instructive to rewrite the final term of 2.20a in an alternative form to give  $K_G$  as

$$K_G = \frac{A}{L} (E - 2S) \frac{x^2}{l^2} + \frac{SA}{L}; \quad S = \sigma \frac{L^2}{l^2}. \quad (2.21)$$

This form introduces an important term called the *second Piola-Kirchhoff* stress  $S$ , which gives the force per unit undeformed area. To achieve this quantity the Cauchy stress  $\sigma$  is multiplied by what will be called *deformation gradient*, that is  $L/l$ .

In Section 2.4 it will be shown that the second Piola-Kirchhoff stress is associated with Green's strain and not with the Cauchy stress, as we assumed in Equation 2.14. Moreover, Equation 2.21 illustrates that the stiffness can be expressed in terms of the initial undeformed configuration or the current deformed configuration.

Equations 2.20 show that the constitutive constant  $E$  has been modified by the current state of stress  $\sigma$  (or  $S$ ). We can see that is a consequence of allowing for geometry changes in the formulation; the  $2\sigma$  term emerges from the derivative of the term  $1/l^2$  in Equation 2.18. Hence whether  $x$  is close to the initial configuration  $X$ , then

$$\left. \begin{array}{l} a \approx A \\ l \approx L \end{array} \right\} \Rightarrow K_L \approx K_G \quad (2.22)$$

Both Equations 2.20a and 2.20b contain a stiffness term  $\sigma a/l$ , equal to  $SA/L$ , which is generally known as the *initial stress stiffness* or *geometric stiffness*. This geometric stiffness is unrelated to the change in cross-sectional area and is purely associated with force changes caused by rigid body rotation [6].

### 2.2.2.3. Continuum Strain Measures

In linear stress-strain analysis the deformation of a continuum body is measured in term of the small strain tensor  $\varepsilon$ . For a simple two-dimensional case  $\varepsilon$  has components  $\varepsilon_{xx}, \varepsilon_{yy}$  and  $\varepsilon_{xy} = \varepsilon_{yx}$ , which are obtained in terms of the  $x$  and  $y$  components of displacement as

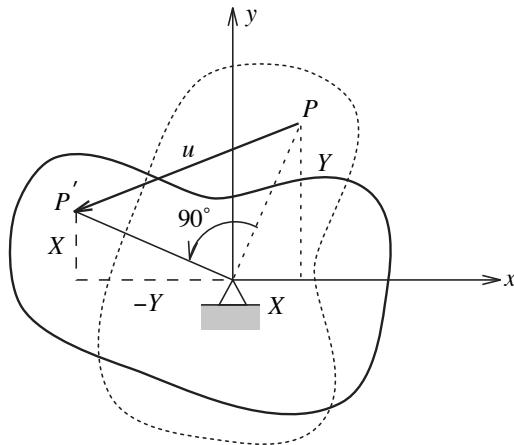
$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x}; \quad (2.23a)$$

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y}; \quad (2.23b)$$

$$\varepsilon_{xy} = \frac{1}{2} \left( \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right). \quad (2.23c)$$

These equations rely on the assumption that the displacements  $u_x$  and  $u_y$  are very small. Therefore, for large strains they are no longer valid. To differentiate between each configuration capital letter are used for initial positions ( $X, Y$ ) while lower case for the current coordinates ( $x, y$ ).

It would be tempting to use Equations 2.23 for the nonlinear case. However, it is easy to demonstrate the results obtained contradict the physical reality. For this purpose we consider a two-dimensional solid undergoing a  $90^\circ$  rotation about the origin as shown in Figure 2.9.



**Figure 2.9:**  $90^\circ$  rotation of a two-dimensional body [6].

The corresponding displacements of any given particle  $P$  are seen from the figure to be

$$u_x = -X - Y; \quad (2.24)$$

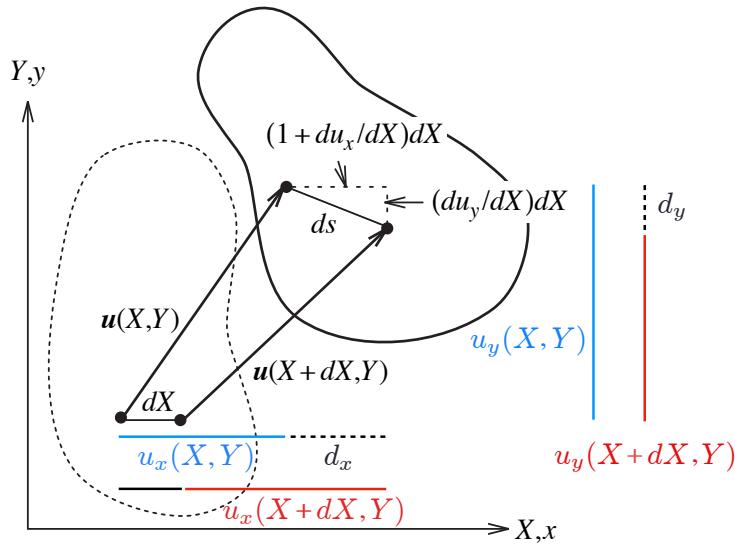
$$u_y = X - Y, \quad (2.25)$$

and the application of Equations 2.23 gives

$$\varepsilon_{xx} = \varepsilon_{yy} = -1; \quad \varepsilon_{xy} = 0. \quad (2.26)$$

These values are incorrect since the solid only experiences rotation and no strain. Consequently, it is necessary to re-establish the definition of strain for a continuum. Although general nonlinear strain measures are discussed extensively in Section 2.3, we can introduce some basic idea by extending the definition of Green's strain in Equation 2.12a to the two-dimensional case.

Consider for this purpose a small elemental segment  $dX$ , which is parallel to  $x$ -axis, deformed to a length  $ds^4$  as shown in Figure 2.10. Thus,  $ds$  can be computed as the sum of the squares of both sides of the triangle  $d_x$  and  $d_y$ .



**Figure 2.10:** General deformation of a two-dimensional body.

The computation of  $d_x$  is done by equalling the deformed and undeformed  $x$  contribution of the element segment  $dX$  as

$$u_x(X, Y) + d_x = dX + u_x(X + dX, Y), \quad (2.27)$$

by using the Taylor series, we approximate  $u_x(X + dX, Y)$  as

$$u_x(X + dX, Y) = u_x(X, Y) + \frac{\partial u_x(X, Y)}{\partial X} dX. \quad (2.28)$$

Then, from Equation 2.27, the length  $d_x$  is isolated as

$$d_x = dX + u_x(X + dX, Y) - u_x(X, Y). \quad (2.29)$$

Substituting Equation 2.28 into the latter expression we lead to

$$d_x = \left( dX + \frac{\partial u_x(X, Y)}{\partial X} dX \right). \quad (2.30)$$

Repeating the process for the  $y$ -axis we obtain the final length of  $ds$  as

$$ds^2 = \left( dX + \frac{\partial u_x}{\partial X} dX \right)^2 + \left( \frac{\partial u_y}{\partial X} dX \right)^2. \quad (2.31)$$

Based on the one-dimensional Green strain (Equation 2.12a), the  $x$  component of the two-dimensional Green strain is

---

<sup>4</sup> $ds^2$  is called the first fundamental form, and it describes the metric properties of a surface.

$$\begin{aligned}
E_{xx} &= \frac{ds^2 - dX^2}{2dX^2} \\
&= \frac{1}{2} \left[ \left( 1 + \frac{\partial u_x}{\partial X} \right)^2 + \left( \frac{\partial u_y}{\partial X} \right)^2 - 1 \right] \\
&= \frac{\partial u_x}{\partial X} + \frac{1}{2} \left[ \left( \frac{\partial u_x}{\partial X} \right)^2 + \left( \frac{\partial u_y}{\partial X} \right)^2 \right]. \tag{2.32}
\end{aligned}$$

Equations for  $E_{yy}$  and the shear strains  $E_{xy} = E_{yx}$  are obtained as

$$E_{yy} = \frac{\partial u_y}{\partial Y} + \frac{1}{2} \left[ \left( \frac{\partial u_x}{\partial Y} \right)^2 + \left( \frac{\partial u_y}{\partial Y} \right)^2 \right]; \tag{2.33}$$

$$E_{xy} = \frac{1}{2} \left( \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \right) + \frac{1}{2} \left( \frac{\partial u_x}{\partial X} \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \frac{\partial u_y}{\partial Y} \right). \tag{2.34}$$

In linear stress-strain analysis, where the displacements are small, the quadratic terms in the above expressions can be ignored and Equations 2.23 are recovered. In addition, it is possible to prove that for the rigid rotation case, the Green strain components are  $E_{xx} = E_{yy} = E_{xy} = 0$ .

## 2.3. Kinematics

Kinematics is the study of motion and deformation without reference. The consideration of finite deformation force us to introduce two different coordinate systems, namely *material* and *spatial* descriptions, which are also associated with the names of *Lagrange* and *Euler* descriptions, respectively.

	Time	Length	Area	Volume	Density	Basis	Coordinate
Material	0	$L$	$A$	$V$	$\rho_0$	$(E_1, E_2, E_3)$	$X$
Spatial	$t$	$l$	$a$	$v$	$\rho$	$(e_1, e_2, e_3)$	$x$

**Table 2.2:** Material and spatial notation.

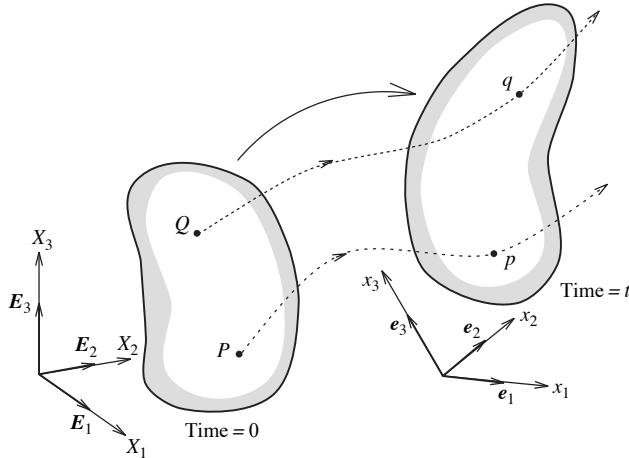
### 2.3.1. The Motion

Figure 2.11 shows the general motion of a deformable body. The body can be seen as an assemblage of material particles that are labeled by coordinates  $X$ , with respect to Cartesian basis  $E_I$  at time  $t = 0$ . The current positions of these particles are designed by the coordinates  $x$ , at time  $t$ , with respect to an alternative Cartesian basis  $e_i$ . For now, the bases  $E_I$  and  $e_i$  will be taken to be coincident.

The motion can be mathematically described by a mapping  $\phi$  between initial and current particle positions as

$$\boldsymbol{x} = \phi(\boldsymbol{X}, t). \quad (2.35)$$

For a fixed value of  $t$ , the above expression represents a mapping between undeformed and deformed bodies. In addition, Equation 2.35 represents, for a fixed  $\boldsymbol{X}$ , the motion or trajectory of a particle as a function of time.



**Figure 2.11:** General motion of a deformable body [7].

### 2.3.2. Material and Spatial Descriptions

In finite deformation analysis a distinction between the coordinate systems is made to describe the behaviour of the body. Relevant quantities, such as density or displacement in our case, can be described in terms of where the body was before the deformation or where it is during this deformation. Nevertheless, the governing equations must obviously refer to where the body is and hence must primarily be formulated using a spatial description [7].

Taking the density as example scalar quantity, the material and spatial descriptions are defined as

- **Material description**,  $\rho = \rho(\boldsymbol{X}, t)$ . The body is at the original coordinate  $\boldsymbol{X}$  in the continuum at time  $t = 0$ . A change in time  $t$  implies that the same material particle  $\boldsymbol{X}$  has a different  $\rho$  value.
- **Spatial description**,  $\rho = \rho(\boldsymbol{x}, t)$ . The quantity is described with respect to the position in space  $\boldsymbol{x}$ , currently occupied by a material particle in the continuum at time  $t$ . A change in time implies that a different density is observed at the same spatial position  $\boldsymbol{x}$ , now probably occupied by a different particle.

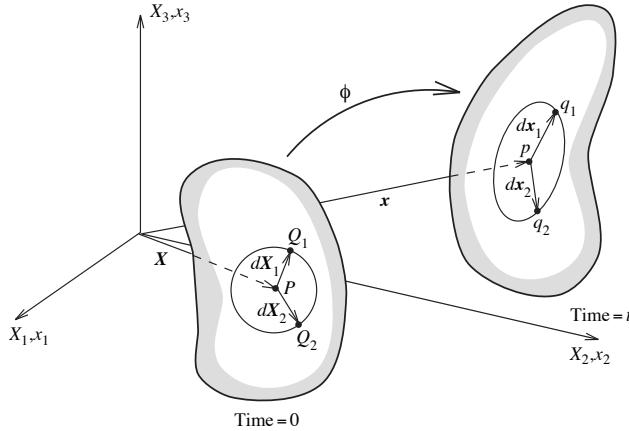
A material description can be obtained from a spatial description as

$$\rho(\boldsymbol{X}, t) = \rho(\phi(\boldsymbol{X}, t), t). \quad (2.36)$$

### 2.3.3. Deformation Gradient

#### 2.3.3.1. Definition

The deformation gradient is a key quantity in finite deformation analysis since it is involved in all equations relating quantities before deformation to corresponding quantities after or during deformation. The deformation gradient tensor  $\mathbf{F}$  enables the relative spatial position of two particles after deformation to be described in terms of their relative material position before deformation. Therefore, it is essential to the description of deformation and hence strain.



**Figure 2.12:** General motion in the neighborhood of a particle [7].

Considering two material particles  $Q_1$  and  $Q_2$  in the neighborhood of a material particle  $P$  (Figure 2.12), their positions are given by the elemental vectors  $d\mathbf{X}_1$  and  $d\mathbf{X}_2$  as

$$d\mathbf{X}_1 = \mathbf{X}_{Q_1} - \mathbf{X}_P; \quad d\mathbf{X}_2 = \mathbf{X}_{Q_2} - \mathbf{X}_P. \quad (2.37)$$

After the deformation, the material particles  $P$ ,  $Q_1$  and  $Q_2$  have deformed to current spatial positions by using the mapping Equation 2.35 as

$$\mathbf{x}_p = \phi(\mathbf{X}_P, t); \quad \mathbf{x}_{q_1} = \phi(\mathbf{X}_{Q_1}, t); \quad \mathbf{x}_{q_2} = \phi(\mathbf{X}_{Q_2}, t), \quad (2.38)$$

and the corresponding elemental vectors in the spatial configuration become

$$d\mathbf{x}_1 = \mathbf{x}_{q_1} - \mathbf{x}_p = \phi(\mathbf{X}_P + d\mathbf{X}_1, t) - \phi(\mathbf{X}_P, t); \quad (2.39)$$

$$d\mathbf{x}_2 = \mathbf{x}_{q_2} - \mathbf{x}_p = \phi(\mathbf{X}_P + d\mathbf{X}_2, t) - \phi(\mathbf{X}_P, t); \quad (2.40)$$

The deformation gradient tensor  $\mathbf{F}$  is defined as the derivative of each component of the deformed configuration vector  $\mathbf{x}$  with respect to each component of the reference vector  $\mathbf{X}$  as

$$\mathbf{F} = \frac{\partial \phi(\mathbf{X}, t)}{\partial \mathbf{X}}. \quad (2.41)$$

Therefore, the elemental vectors  $d\mathbf{x}_1$  and  $d\mathbf{x}_2$  can be obtained after a Taylor series expansion in terms of  $d\mathbf{X}_1$  and  $d\mathbf{X}_2$  as

$$d\mathbf{x}_1 = \mathbf{F} d\mathbf{X}_1; \quad d\mathbf{x}_2 = \mathbf{F} d\mathbf{X}_2. \quad (2.42)$$

As  $\mathbf{F}$  transforms vectors in the reference configuration into vectors in the current (or updated) configuration it is said to be a two-point tensor.

Sometimes, in literature, the motion of a particle is expressed as  $\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$  which permits the deformation gradient tensor to be written as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \quad (2.43)$$

In matrix form, for 3D problems, is expressed as

$$\mathbf{F} = \begin{pmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{pmatrix}. \quad (2.44)$$

Moreover, an indicial tensor notation can be used to express the deformation gradient tensor. This type of notation allows to translate equations easily into a computer. Thus, in indicial notation, the deformation gradient tensor is written as

$$\mathbf{F} = \sum_{i,I=1}^3 F_{iI} \mathbf{e}_i \otimes \mathbf{E}_I; \quad F_{iI} = \frac{\partial x_i}{\partial X_I}; \quad i, I = 1, 2, 3 \quad (2.45)$$

where lowercase indices refer to current Cartesian coordinates, whereas uppercase indices refer to initial Cartesian coordinates.

### 2.3.3.2. Push Forward & Pull Back

Considering a single elemental material vector  $d\mathbf{X}$ , the corresponding vector  $d\mathbf{x}$  in the spatial configuration is written as  $d\mathbf{x} = \mathbf{F} d\mathbf{X}$ . However, the opposite operation could be done by using the inverse of  $\mathbf{F}$  as

$$d\mathbf{X} = \mathbf{F}^{-1} d\mathbf{x}, \quad (2.46)$$

where the inverse of  $\mathbf{F}$  is

$$\mathbf{F}^{-1} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}}; \quad \mathbf{F}^{-1} = \sum_{I,i=1}^3 \frac{\partial X_I}{\partial x_i} \mathbf{E}_I \otimes \mathbf{e}_i. \quad (2.47)$$

This relationship between quantities in the material and spatial configurations is expressed in terms of the general concepts of *push forward* and *pull back*. For example,  $d\mathbf{x}$  is said to be the push forward of  $d\mathbf{X}$ . Both push forward and pull back operations can be represented, respectively, as

$$d\mathbf{x} = \phi_*[d\mathbf{X}] = \mathbf{F} d\mathbf{X}; \quad (2.48)$$

$$d\mathbf{X} = \phi_*^{-1}[d\mathbf{x}] = \mathbf{F}^{-1} d\mathbf{x}, \quad (2.49)$$

where  $\phi_*$  represents the push-forward and pull-back operations, respectively.

### 2.3.3.3. Rigid Body Motion

The displacement  $\mathbf{u}$  is defined as

$$\mathbf{u} = \mathbf{x} - \mathbf{X}. \quad (2.50)$$

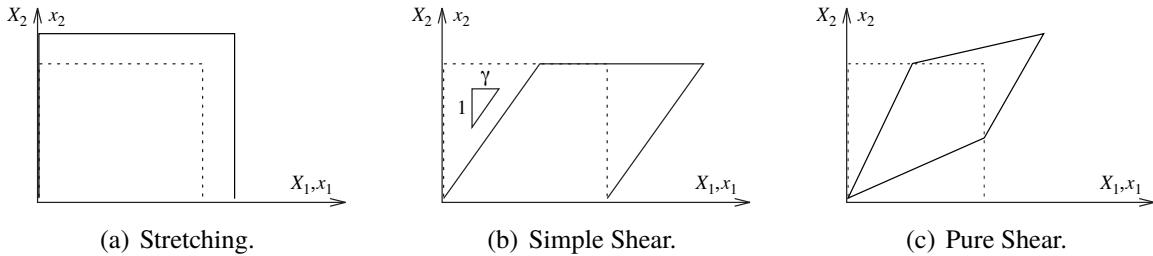
This leads to  $\mathbf{x} = \mathbf{X} + \mathbf{u}$ , and thus the deformation gradient can be expressed as

$$\mathbf{F} = \frac{\partial}{\partial \mathbf{X}}(\mathbf{X} + \mathbf{u}) = \frac{\partial \mathbf{X}}{\partial \mathbf{X}} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}. \quad (2.51)$$

Rigid body displacements do not appear in the deformation gradient since they do not contribute to stress and strain. From Equation 2.51 the following important outcome is deduced: whether there is no deformation  $\mathbf{F} = \mathbf{I}$ . This is also an indicative of a lack of rigid body rotations. However, as it will be discussed in Section 2.3.5, rotations alter the value of  $\mathbf{F}$  so it is no longer  $\mathbf{I}$ .

### 2.3.3.4. Simple Deformations

Three two-dimensional simple cases are depicted below to see how deformation gradient tensor looks like.



**Figure 2.13:** Simple deformations of a two-dimensional body.

Figure 2.13(a) is stretched in the  $x$  and  $y$  directions. The motion equations describe a 100% elongation in the  $x$ -direction while it suffers a 50% elongation in the  $y$ -direction. Figure 2.13(b) represents a simple shear. The non-zero off-diagonal value  $\gamma$  reflects shear. Pure shear, in Figure 2.13(c), has a deformation gradient with non-zero off-diagonal values meaning that shearing is present. Additionally, the fact that  $\mathbf{F}$  is symmetric reflects that there is no net rotation.

Case	Stretching	Simple Shear	Pure Shear
Motion	$x_1 = 2X_1$ $x_2 = 1.5X_2$	$x_1 = X_1 + \gamma X_2$ $x_2 = X_2$	$x_1 = X_1 + 0.5X_2$ $x_2 = 0.5X_1 + X_2$
Deformation Gradient	$\mathbf{F} = \begin{bmatrix} 2 & 0 \\ 0 & 1.5 \end{bmatrix}$	$\mathbf{F} = \begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix}$	$\mathbf{F} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$

**Table 2.3:** Deformation gradient examples.

### 2.3.4. Deformation and Strain

#### 2.3.4.1. Metric Tensor

The previous definition of the deformation gradient tensor allow us to redefine the strain. The line element or first fundamental form  $ds$  is invariant since the distance between two neighboring points is independent of the coordinate system. This line element is defined by the inner product of the elemental vector in the spatial configuration as

$$ds^2 = d\mathbf{x} \cdot d\mathbf{x}, \quad (2.52)$$

by using the definition of push forward,

$$ds^2 = \mathbf{F}d\mathbf{X} \cdot \mathbf{F}d\mathbf{X} = d\mathbf{X}^T \mathbf{F}^T d\mathbf{X}. \quad (2.53)$$

The first fundamental form can be expressed in a more general form by using the fundamental metric tensor  $g$  as

$$ds^2 = g(d\mathbf{X}, d\mathbf{X}), \quad (2.54)$$

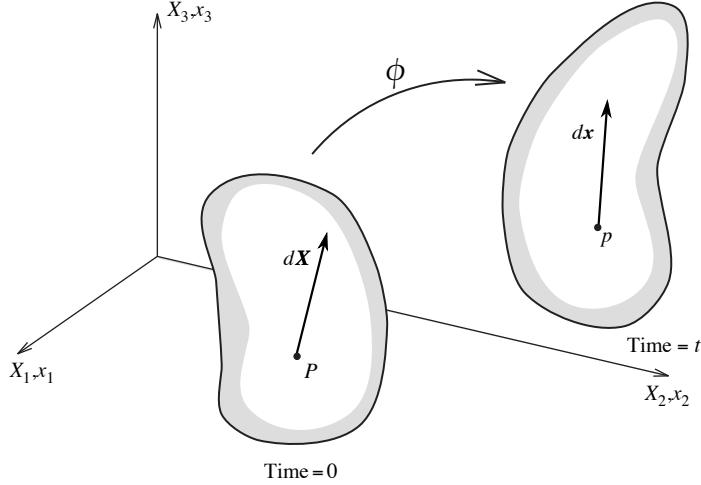
where the metric tensor  $g$  is a type of function which takes a pair of vectors at a point of a surface and produces a real number scalar. Whether this metric tensor assigns a positive value  $g(d\mathbf{X}, d\mathbf{X}) > 0$ , its matrix representation is said to be *positive-definite*.

#### 2.3.4.2. Stretch

The ratio of the length of the spatial vector  $d\mathbf{x}$  to the length of the corresponding reference vector  $d\mathbf{X}$  is called stretch

$$\lambda = \frac{\|d\mathbf{x}\|}{\|d\mathbf{X}\|} = \frac{\sqrt{d\mathbf{x} \cdot d\mathbf{x}}}{\sqrt{d\mathbf{X} \cdot d\mathbf{X}}} > 0. \quad (2.55)$$

By choosing  $\|d\mathbf{X}\| = 1$  as the reference value, the stretch can be expressed as



**Figure 2.14:** Change in length.

$$\begin{aligned}
 \lambda &= \sqrt{d\mathbf{x} \cdot d\mathbf{x}} \\
 &= \sqrt{(\mathbf{F}d\mathbf{X}) \cdot (\mathbf{F}d\mathbf{X})} \\
 &= \sqrt{d\mathbf{X} \cdot (\mathbf{F}^T \mathbf{F})d\mathbf{X}} \\
 &= \sqrt{d\mathbf{X} \cdot \mathbf{C} d\mathbf{X}} \\
 &= \|d\mathbf{X}\|_C,
 \end{aligned} \tag{2.56}$$

where  $\mathbf{C}$  is the *right Cauchy-Green deformation tensor* and it is defined in terms of the gradient deformation tensor as

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}. \tag{2.57}$$

Note that the tensor  $\mathbf{C}$  operates on the material vector  $d\mathbf{X}$ , and consequently  $\mathbf{C}$  is called a material tensor quantity. The right Cauchy-Green tensor is symmetric and positive-definite,

$$\mathbf{C} = \mathbf{C}^T = (\mathbf{F}^T \mathbf{F})^T = \mathbf{F}^T \mathbf{F} \quad \text{and} \quad d\mathbf{X} \cdot \mathbf{C} d\mathbf{X} \geq 0, \tag{2.58}$$

where  $d\mathbf{X} \cdot \mathbf{C} d\mathbf{X}$  is the general form of the inner product. Thus, the eigenvalues of  $\mathbf{C}$  are  $\lambda_\alpha^2$ , which are positive and real numbers. The stretch formulae  $\lambda = \|d\mathbf{X}\|_C$  and  $\lambda = \|d\mathbf{x}\|_1$  settle the Lagrangian and Eulerian descriptions.

Alternatively, instead of imposing the value of the material vector, a Eulerian (spatial) approach is done by imposing  $\|d\mathbf{x}\|_1 = 1$ . Then the inverse of the stretch is

$$\begin{aligned}
\lambda^{-1} &= \sqrt{d\mathbf{X} \cdot d\mathbf{X}} \\
&= \sqrt{(\mathbf{F}^{-1}d\mathbf{x}) \cdot (\mathbf{F}^{-1}d\mathbf{x})} \\
&= \sqrt{d\mathbf{x} \cdot (\mathbf{F}^{-T}\mathbf{F}^{-1})d\mathbf{x}} \\
&= \sqrt{d\mathbf{x} \cdot \mathbf{b}^{-1}d\mathbf{x}} \\
&= \|d\mathbf{x}\|_{\mathbf{b}^{-1}}, 
\end{aligned} \tag{2.59}$$

the left Cauchy-Green tensor  $\mathbf{b}$  is introduced, and defined as

$$\mathbf{b} = \mathbf{F}\mathbf{F}^T. \tag{2.60}$$

From this definition it is evident that  $\mathbf{b}$  is a symmetric and positive-definite tensor. From Equations 2.57 and 2.60 we observe that  $\mathbf{C}$  and  $\mathbf{b}$  act as metric tensor in the Eulerian and Lagrangian descriptions of the length deformation.

#### 2.3.4.3. Strain Tensors

Once the stretch is defined in both descriptions by using the adequate metric, it is now possible to derive the strain tensors. Recalling Equation 2.12a, the one-dimensional Green strain can be defined in terms of the stretch  $\lambda$  as

$$\varepsilon_G = \frac{1}{2}(\lambda^2 - 1). \tag{2.61}$$

A strain measure simply compares the difference between metrics. Therefore, the Lagrangian strain will compute the difference between material and spatial metrics. The spatial measure compares the first fundamental forms of both configurations, which are the squared lengths, in this manner  $\|d\mathbf{x}\|_1^2 = \|d\mathbf{X}\|_C^2$ . In the Lagrangian configuration the length of its reference vector is set to be the unity  $\|d\mathbf{X}\|_1 = 1$ . Thus, inserting these definitions into Equation 2.61 gives

$$\begin{aligned}
\varepsilon_G &= \frac{1}{2}(\|d\mathbf{X}\|_C^2 - \|d\mathbf{X}\|_1^2) \\
&= \frac{1}{2}(d\mathbf{X} \cdot \mathbf{C}d\mathbf{X} - d\mathbf{X} \cdot \mathbf{1}d\mathbf{X}) \\
&= d\mathbf{X} \cdot \frac{1}{2}(\mathbf{C} - \mathbf{I})d\mathbf{X} \\
&= d\mathbf{X} \cdot \mathbf{E}d\mathbf{X}.
\end{aligned} \tag{2.62}$$

From this expression the Green-Lagrange strain tensor  $\mathbf{E}$  is then defined as

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}). \tag{2.63}$$

Similarly, an analogous strain measure can be expressed in terms of the spatial reference by using the inverse of the stretch. Using the definition of Almansi strain 2.12b and imposing  $\|d\mathbf{x}\|_1^2 = 1$  and  $\|d\mathbf{x}\|_{b^{-1}}^2 = \lambda^{-2}$  we accomplish the *Eulerian* or *Almansi strain tensor*  $\mathbf{e}$  as

$$\begin{aligned}\varepsilon_A &= \frac{1}{2}(\|d\mathbf{x}\|_1^2 - \|d\mathbf{x}\|_{b^{-1}}^2) \\ &= \frac{1}{2}(d\mathbf{x} \cdot 1 d\mathbf{x} - d\mathbf{x} \cdot \mathbf{b}^{-1} d\mathbf{x}) \\ &= d\mathbf{x} \cdot \frac{1}{2}(1 - \mathbf{b}^{-1}) d\mathbf{x} \\ &= d\mathbf{x} \cdot \mathbf{e}^{-1} d\mathbf{x},\end{aligned}\tag{2.64}$$

where  $\mathbf{e}$  is defined as follows

$$\mathbf{e} = \frac{1}{2}(\mathbf{I} - \mathbf{b}^{-1}).\tag{2.65}$$

Note that the tensor notation forces the replacement of 1 by the identity matrix  $\mathbf{I}$ . These strain measures can be related through the push-forward and pull-back operations in the following way

$$\mathbf{e} = \phi_*[\mathbf{E}] = \mathbf{F}^{-T} \mathbf{E} \mathbf{F}^{-1}\tag{2.66}$$

$$\mathbf{E} = \phi_*^{-1}[\mathbf{e}] = \mathbf{F}^T \mathbf{e} \mathbf{F}.\tag{2.67}$$

### 2.3.5. Polar Decomposition

The gradient deformation tensor  $\mathbf{F}$  transforms any material vector  $d\mathbf{X}$  into the corresponding spatial vector  $d\mathbf{x}$ . To study the key role of  $\mathbf{F}$  it is disclosed into two components: stretch and rotation. Therefore, the gradient deformation tensor  $\mathbf{F}$  can be viewed as the product of a *rotation tensor*  $\mathbf{R}$  times the *stretch tensor*  $\mathbf{U}$  to define the polar decomposition as

$$\mathbf{F} = \mathbf{R} \mathbf{U}.\tag{2.68}$$

By introducing this expression into the right-Cauchy strain tensor definition we obtain

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} = \mathbf{U}^T \mathbf{R}^T \mathbf{R} \mathbf{U}.\tag{2.69}$$

Since the rotation tensor is orthogonal,  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ , and choosing the stretch tensor to be symmetric,  $\mathbf{U} = \mathbf{U}^T$ , we obtain a second definition of the right Cauchy-Green deformation tensor as

$$\mathbf{C} = \mathbf{U}^2.\tag{2.70}$$

To obtain  $\mathbf{U}$  it is necessary to evaluate the principal directions of  $\mathbf{C}$ , denoted by the eigenvector triad  $\{\mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3\}$ , and their corresponding eigenvalues  $\lambda_{\alpha=1,2,3}^2$ , which enable  $\mathbf{C}$  to be expressed as

$$\mathbf{C} = \sum_{\alpha=1}^3 \lambda_{\alpha}^2 \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}, \quad (2.71)$$

where the triad  $\{\mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3\}$  are orthogonal unit vectors because of the symmetry of  $\mathbf{C}$  [7]. Then, the stretch tensor is computed as

$$\mathbf{U} = \sum_{\alpha=1}^3 \lambda_{\alpha} \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}. \quad (2.72)$$

After computing the stretch tensor  $\mathbf{U}$ , the rotation tensor  $\mathbf{R}$  is simply

$$\mathbf{R} = \mathbf{F} \mathbf{U}^{-1}. \quad (2.73)$$

Therefore, when a material vector  $d\mathbf{X}$  is transformed into a spatial vector  $d\mathbf{x}$  by means of  $\mathbf{F}$  its movement can be split in two tensors,

$$d\mathbf{x} = \mathbf{F} d\mathbf{X} = \mathbf{R}(\mathbf{U} d\mathbf{X}), \quad (2.74)$$

where  $d\mathbf{X}$  first is stretched obtaining  $\mathbf{U} d\mathbf{X}$ , and then is rotated to the spatial configuration  $\mathbf{R}$ . Thus,  $\mathbf{R}$  is the responsible of changing between the two configurations.

The analogous operation can be done to decompose  $\mathbf{F}$  in the updated configuration by the same rotation tensor  $\mathbf{R}$  and the stretch in the spatial configuration  $\mathbf{V}$  as

$$\mathbf{F} = \mathbf{V} \mathbf{R}. \quad (2.75)$$

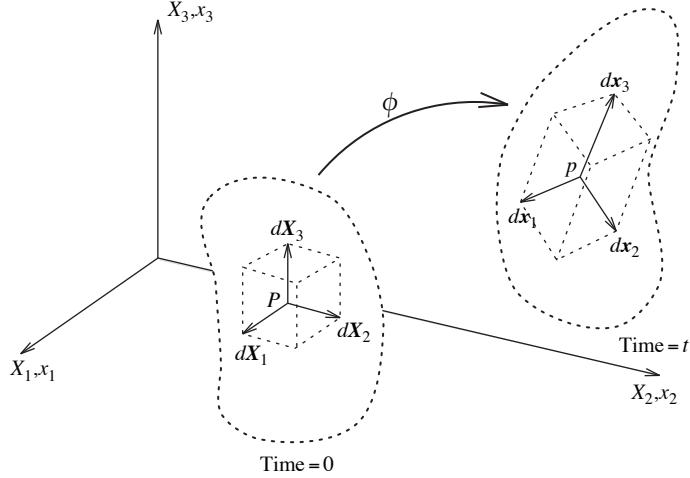
Now the operations are exchanged, first  $d\mathbf{X}$  is rotated into the spatial configuration and then it is stretched to give  $d\mathbf{x}$ . In a similar way the left Cauchy-Green tensor and the stretch can be expressed in terms of the spatial stretch, and the eigenvalues, respectively as

$$\mathbf{b} = \mathbf{F} \mathbf{F}^T = (\mathbf{V} \mathbf{R})(\mathbf{R}^T \mathbf{V}) = \mathbf{V}^2; \quad (2.76)$$

$$\mathbf{V} = \sum_{\alpha=1}^3 \lambda_{\alpha} \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha}. \quad (2.77)$$

### 2.3.6. Volume & Area Change

Considering an infinitesimal volume element in the material configuration with edges parallel to the Cartesian axes given by



**Figure 2.15:** Volume change.

$$\begin{aligned} d\mathbf{X}_1 &= dX_1 \mathbf{E}_1; \\ d\mathbf{X}_2 &= dX_2 \mathbf{E}_2; \\ d\mathbf{X}_3 &= dX_3 \mathbf{E}_3, \end{aligned} \quad (2.78)$$

where  $\mathbf{E}_1$ ,  $\mathbf{E}_2$  and  $\mathbf{E}_3$  are the orthogonal unit vectors (Figure 2.15). The elemental material volume  $dV$  is defined by

$$dV = dX_1 dX_2 dX_3. \quad (2.79)$$

To compute the deformed volume  $dv$  in the spatial configuration it will be necessary to obtain the spatial vectors by pushing forward the material vectors. The volume  $dv$  is defined by the triple product as

$$dv = d\mathbf{x}_1 \cdot (d\mathbf{x}_2 \times d\mathbf{x}_3). \quad (2.80)$$

Note that this triple product is indeed the determinant of  $\mathbf{F}$ , and gives the volume change in terms of the Jacobian as

$$dv = J dV; \quad J = |\mathbf{F}|. \quad (2.81)$$

Considering an element of area in the material configuration  $d\mathbf{A} = dA \mathbf{N}$  which its deformed version is  $d\mathbf{a} = da \mathbf{n}$  as shown in Figure 2.16. In order to implicate these two area vectors, an arbitrary material vector  $d\mathbf{L}$  is introduced. Its spatial representation is denoted by  $dl$ . The corresponding initial and current volumes are

$$dV = d\mathbf{L} \cdot d\mathbf{A}; \quad (2.82)$$

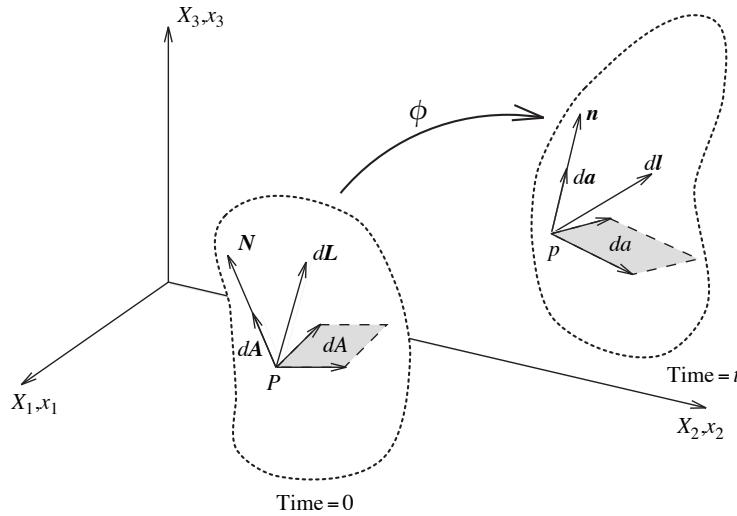
$$dv = dl \cdot da. \quad (2.83)$$

Knowing the Jacobian  $J$  is the relation between the current and the initial volumes, and recalling that  $dl = \mathbf{F} d\mathbf{L}$  leads to

$$J d\mathbf{L} \cdot d\mathbf{A} = (\mathbf{F} d\mathbf{L}) \cdot d\mathbf{a}. \quad (2.84)$$

The fact that this expression is valid for any length vector enables the elements of area to be related as

$$d\mathbf{a} = J \mathbf{F}^{-T} d\mathbf{A}. \quad (2.85)$$



**Figure 2.16:** Area change.

### 2.3.7. Linearized Kinematics

The strain quantities defined in the previous sections are nonlinear expressions due to its dependency of the motion  $\phi$ , and thus will lead to nonlinear governing equations. Before using the Newton-Raphson scheme to solve these equations a linearization process must be done. Strain quantities are linearized with respect to small changes in motion.

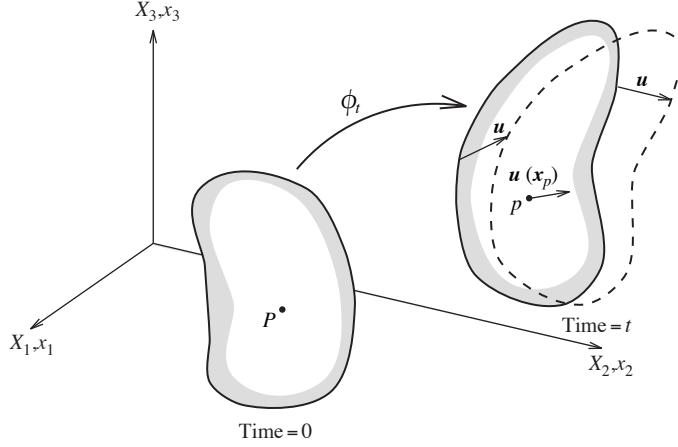
#### 2.3.7.1. Linearization Approach

Considering a nonlinear equation  $f(x) = 0$  and an initial guess of the solution  $x_0$ , the function  $f(x)$  can be expressed in the neighborhood of  $x_0$  using Taylor's series as

$$f(x) = f(x_0) + \frac{df}{dx}\Big|_{x_0} (x - x_0) + \frac{1}{2} \frac{d^2 f}{dx^2}\Big|_{x_0} (x - x_0)^2 + \dots \quad (2.86)$$

expressing the  $x$  increment as  $u = (x - x_0)$

$$f(x_0 + u) = f(x_0) + \frac{df}{dx}\Big|_{x_0} u + \frac{1}{2} \frac{d^2 f}{dx^2}\Big|_{x_0} u^2 + \dots \quad (2.87)$$



**Figure 2.17:** Linearized kinematics.

truncating the Taylor's expression to give a linear function in  $u$  as

$$f(x_0 + u) \approx f(x_0) + \frac{df}{dx} \Big|_{x_0} u. \quad (2.88)$$

Thus, the linearized increment in  $f(x)$  at  $x_0$  with respect to  $u$  is generally expressed as

$$Df(x_0)[u] = \frac{df}{dx} \Big|_{x_0} u \approx f(x_0 + u) - f(x_0). \quad (2.89)$$

### 2.3.7.2. Linearized Deformation Gradient

Consider a small displacement  $\mathbf{u}(x)$  from the current configuration  $\mathbf{x} = \phi_t(\mathbf{X}) = \phi(\mathbf{X}, t)$  as shown in Figure 2.17. The deformation gradient  $\mathbf{F}$  can be linearized, using the Gâteaux derivative<sup>5</sup>, in the direction of  $\mathbf{u}$  as

$$\begin{aligned} D\mathbf{F}(\phi_t)[\mathbf{u}] &= \frac{d}{d\epsilon} \Big|_{\epsilon=0} \mathbf{F}(\phi_t + \epsilon\mathbf{u}) \\ &= \frac{d}{d\epsilon} \Big|_{\epsilon=0} \frac{\partial(\phi_t + \epsilon\mathbf{u})}{\partial\mathbf{X}} \\ &= \frac{d}{d\epsilon} \Big|_{\epsilon=0} \left( \frac{\partial\phi_t}{\partial\mathbf{X}} + \epsilon \frac{\partial\mathbf{u}}{\partial\mathbf{X}} \right) \\ &= \frac{\partial\mathbf{u}}{\partial\mathbf{X}} \\ &= (\nabla\mathbf{u})\mathbf{F}. \end{aligned} \quad (2.90)$$

<sup>5</sup>The Gâteaux differential is a generalization of directional derivative, and it is defined as  $dF(\phi)[u]$  of  $F$  at  $\phi$  in the direction  $u$  is defined as  $dF(\phi)[u] = \lim_{\epsilon \rightarrow 0} \frac{F(\phi + \epsilon u) - F(\phi)}{\epsilon} = \frac{d}{d\epsilon} F(\phi + \epsilon u) \Big|_{\epsilon=0}$ .

If  $\mathbf{u}$  is given as a function of the initial position of the body particles  $\mathbf{X}$ , then

$$D\mathbf{F}[\mathbf{u}] = \frac{\partial \mathbf{u}(\mathbf{X})}{\partial \mathbf{X}} = \nabla_0 \mathbf{u}, \quad (2.91)$$

where  $\nabla_0$  indicates the gradient with respect to the coordinates at the initial configuration.

### 2.3.7.3. Linearized Strain

Using Equation 2.90 and the product rule<sup>6</sup>, the Lagrangian strain can be linearized in the direction  $\mathbf{u}$  as

$$\begin{aligned} D\mathbf{E}[\mathbf{u}] &= \frac{1}{2}(\mathbf{F}^T D\mathbf{F}[\mathbf{u}] + D\mathbf{F}^T[\mathbf{u}\mathbf{F}]) \\ &= \frac{1}{2}(\mathbf{F}^T \nabla \mathbf{u} \mathbf{F} + \mathbf{F}^T (\nabla \mathbf{u})^T \mathbf{F}) \\ &= \frac{1}{2} \mathbf{F}^T (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \mathbf{F}. \end{aligned} \quad (2.92)$$

Note that half the tensor inside the parenthesis is the small strain tensor  $\boldsymbol{\varepsilon}$ , and therefore,  $D\mathbf{E}[\mathbf{u}]$  can be interpreted as the pull back of the small strain tensor  $\boldsymbol{\varepsilon}$  as [7]

$$D\mathbf{E}[\mathbf{u}] = \phi_*^{-1}[\boldsymbol{\varepsilon}] = \mathbf{F}^T \boldsymbol{\varepsilon} \mathbf{F}. \quad (2.93)$$

If the linearization of  $\mathbf{E}$  is performed at the initial configuration ( $\mathbf{x} = \mathbf{X}$ ), and therefore  $\mathbf{F} = \mathbf{I}$ , then

$$D\mathbf{E}_0[\mathbf{u}] = \boldsymbol{\varepsilon}. \quad (2.94)$$

Similarly, the right and left Cauchy-Green deformation tensors can be linearized to give

$$D\mathbf{C}[\mathbf{u}] = 2\mathbf{F}^T \boldsymbol{\varepsilon} \mathbf{F}; \quad (2.95)$$

$$D\mathbf{b}[\mathbf{u}] = (\nabla \mathbf{u}) \mathbf{b} + \mathbf{b} (\nabla \mathbf{u})^T. \quad (2.96)$$

### 2.3.7.4. Linearized Volume Change

The volume change is given by the Jacobian  $J = dv/dV$ . Using the chain rule<sup>7</sup>, the directional derivative of  $J$  with respect to an increment  $\mathbf{u}$  in the updated configuration is

$$D J[\mathbf{u}] = D \det(\mathbf{F})[D\mathbf{F}[\mathbf{u}]], \quad (2.97)$$

where the directional derivative of the determinant can be expressed as

---

<sup>6</sup>The product rule property of the directional derivative, if  $F(x) = F_1(x) \cdot F_2(x)$ , is then  $DF(x_0)[u] = DF_1(x_0)[u] \cdot F_2(x_0) + F_1(x_0) \cdot DF_2(x_0)[u]$ .

<sup>7</sup>The chain rule, if  $F(x) = F_1(F_2(x))$ , is  $DF(x_0)[u] = DF_1(F_2(x_0))[DF_2(x_0)[u]]$ .

$$D J[\mathbf{u}] = J \operatorname{tr} \left( \mathbf{F}^{-1} \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right). \quad (2.98)$$

Substituting the linearization of  $\mathbf{F}$  in the latter equation gives

$$\begin{aligned} D J[\mathbf{u}] &= J \operatorname{tr} \nabla \mathbf{u} \\ &= J \nabla \cdot \mathbf{u}. \end{aligned} \quad (2.99)$$

### 2.3.8. Velocity and Material Time Derivatives

#### 2.3.8.1. Velocity

Many nonlinear processes are time-dependent, and consequently it is necessary to consider velocity and material time derivatives of various quantities. However, even if the process is not time-dependent it is convenient to establish the equilibrium equations in terms of the virtual and associated virtual time-dependant quantities. Considering the usual motion of the body given by Equation 2.35 as

$$\mathbf{x} = \phi(\mathbf{X}, t), \quad (2.100)$$

from which the velocity of a particle is defined as the time derivative of  $\phi$  as

$$\mathbf{v}(\mathbf{X}, t) = \frac{\partial \phi(\mathbf{X}, t)}{\partial t}. \quad (2.101)$$

Although the latter equation is expressed in terms of the material coordinates of the particle  $\mathbf{X}$ , the velocity vector is a spatial quantity. Therefore, a more proper definition of velocity is

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v}(\phi^{-1}(\mathbf{x}, t), t). \quad (2.102)$$

#### 2.3.8.2. Material Time Derivative

Given a general scalar or tensor quantity  $\mathbf{g}$ , expressed in terms of the material coordinates  $\mathbf{X}$ , the time derivative of  $g(\mathbf{X}, t)$  denoted by  $\dot{\mathbf{g}}$  is

$$\dot{\mathbf{g}} = \frac{d\mathbf{g}}{dt} = \frac{\partial \mathbf{g}(\mathbf{X}, t)}{\partial t}, \quad (2.103)$$

this expression measures the change in  $\mathbf{g}$  with a specific particle initially located at  $\mathbf{X}$ , and it is known as the *material time derivative* of  $\mathbf{g}$ . The material derivative of a spatial quantity is more complicated to establish due to change in position of the specific particle as time progresses. Nevertheless, the material time derivative can be obtained using the following expression

$$\dot{\mathbf{g}}(\mathbf{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{g}(\phi(\mathbf{X}, t + \Delta t), t + \Delta t) - \mathbf{g}(\phi(\mathbf{X}, t), t)}{\Delta t}. \quad (2.104)$$

This equation illustrates that  $\mathbf{g}$  changes in time due to

- a change in time but with the particle remaining in the same spatial position.
- the change in spatial position of the specific particle.

Using the chain rule, Equation 2.103 gives the material derivative of  $\mathbf{g}(\mathbf{x}, t)$  as

$$\dot{\mathbf{g}}(\mathbf{x}, t) = \frac{\partial \mathbf{g}(\mathbf{x}, t)}{\partial t} + \frac{\partial \mathbf{g}(\mathbf{x}, t)}{\partial \mathbf{x}} \frac{\partial \phi(\mathbf{X}, t)}{\partial t} = \frac{\partial \mathbf{g}(\mathbf{x}, t)}{\partial t} + (\nabla \mathbf{g}) \mathbf{v}, \quad (2.105)$$

where the second term, involving the particle velocity, is known as the *convective derivative*.

### 2.3.8.3. Velocity Gradient

We have defined the velocity as a spatial vector in Equation 2.102. Thus, the derivative of this expression with respect to the spatial coordinates defines the *velocity gradient tensor*  $\mathbf{l}$  as

$$\mathbf{l} = \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial \mathbf{x}} = \nabla \mathbf{v}. \quad (2.106)$$

Additionally, the time derivative of the deformation gradient can be written as

$$\dot{\mathbf{F}} = \frac{d}{dt} \left( \frac{\partial \phi}{\partial \mathbf{X}} \right) = \frac{\partial}{\partial \mathbf{X}} \left( \frac{\partial \phi}{\partial t} \right) = \nabla_0 \mathbf{v}. \quad (2.107)$$

It is possible to define the time derivative of the deformation gradient tensor in terms of tensor  $\mathbf{l}$  as

$$\dot{\mathbf{F}} = \frac{\partial \mathbf{v}}{\partial \mathbf{X}} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{\partial \phi}{\partial \mathbf{X}} = \mathbf{l} \mathbf{F}, \quad (2.108)$$

from which  $\mathbf{l}$  can be rewritten as

$$\mathbf{l} = \dot{\mathbf{F}} \mathbf{F}^{-1}. \quad (2.109)$$

### 2.3.9. Rate of Deformation

Considering two initial elemental vectors  $d\mathbf{X}_1$  and  $d\mathbf{X}_2$  and their corresponding pushed forward spatial counterparts  $d\mathbf{x}_1$  and  $d\mathbf{x}_2$ , the strain can be defined and measured as the change in the scalar product as

$$d\mathbf{x}_1 \cdot d\mathbf{x}_2 = d\mathbf{X}_1 \cdot \mathbf{C} d\mathbf{X}_2. \quad (2.110)$$

Differentiating this expression with respect to time and recalling the relationship between the Lagrangian strain tensor  $\mathbf{E}$  and the right Cauchy-Green tensor as  $2\mathbf{E} = (\mathbf{C} - \mathbf{I})$ ; the current rate of change of the scalar product takes the following form

$$\frac{d}{dt}(d\mathbf{x}_1 \cdot d\mathbf{x}_2) = d\mathbf{X}_1 \cdot \dot{\mathbf{C}}d\mathbf{X}_2 = 2d\mathbf{X}_1 \cdot \dot{\mathbf{E}}d\mathbf{X}_2, \quad (2.111)$$

where  $\dot{\mathbf{E}}$  is the derivative with respect to time of the Lagrangian strain tensor, and it is known as the *material strain rate tensor* and can be expressed in terms of  $\dot{\mathbf{F}}$  as

$$\dot{\mathbf{E}} = \frac{1}{2}\dot{\mathbf{C}} = \frac{1}{2}(\dot{\mathbf{F}}^T \mathbf{F} + \mathbf{F}^T \dot{\mathbf{F}}). \quad (2.112)$$

The strain rate tensor  $\dot{\mathbf{E}}$  gives the current rate of change of the scalar product in terms of the initial elemental vectors. Furthermore, it is possible to express the same rate in terms of the current spatial vectors by isolating the material vector of the push-forward definition as

$$d\mathbf{X}_1 = \mathbf{F}^{-1} d\mathbf{x}_1; \quad d\mathbf{X}_2 = \mathbf{F}^{-1} d\mathbf{x}_2. \quad (2.113)$$

Introducing these expressions into Equation 2.111, gives the rate of change of the scalar product in terms of  $d\mathbf{x}_1$  and  $d\mathbf{x}_2$  as

$$\frac{1}{2} \frac{d}{dt}(d\mathbf{x}_1 \cdot d\mathbf{x}_2) = d\mathbf{x}_1 \cdot (\mathbf{F}^{-T} \dot{\mathbf{E}} \mathbf{F}^{-1}) d\mathbf{x}_2. \quad (2.114)$$

The tensor in the expression on the right-hand side is the pushed forward spatial counterpart of  $\dot{\mathbf{E}}$  and is known as the *rate of deformation tensor*  $\mathbf{d}$  given as

$$\mathbf{d} = \phi_*[\dot{\mathbf{E}}] = \mathbf{F}^{-T} \dot{\mathbf{E}} \mathbf{F}^{-1}; \quad \dot{\mathbf{E}} = \phi_*^{-1}[\mathbf{d}] = \mathbf{F}^T \mathbf{d} \mathbf{F}. \quad (2.115)$$

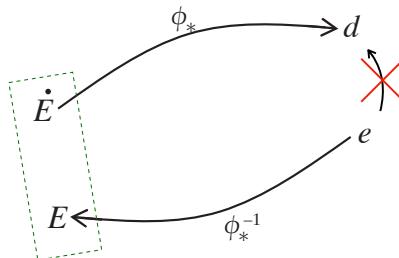
Introducing Equation 2.112 for  $\dot{\mathbf{E}}$  and Equation 2.108 for  $\dot{\mathbf{F}}$  to give the tensor  $\mathbf{d}$  as the symmetric part of  $\mathbf{l}$  as

$$\mathbf{d} = \frac{1}{2}(\mathbf{l} + \mathbf{l}^T). \quad (2.116)$$

It is important to emphasize the spatial rate of deformation tensor  $\mathbf{d}$  is not the material derivative of the Almansi or spatial strain tensor  $\mathbf{e}$ . Instead,  $\mathbf{d}$  is the push forward of  $\dot{\mathbf{E}}$ , which is the derivative with respect to time of the pull back of  $\mathbf{e}$ , as indicated in Equation 2.117. The differentiation with respect to time the spatial strain tensor  $\mathbf{e}$  in order to obtain the deformation rate  $\mathbf{d}$  is erroneous.

$$\mathbf{d} = \phi_* \left[ \frac{d}{dt}(\phi_*^{-1}[\mathbf{e}]) \right]. \quad (2.117)$$

This operation is illustrated in Figure 2.18 and it is known as the *Lie derivative* of a tensor quantity over the mapping  $\phi$  and is generally expressed as



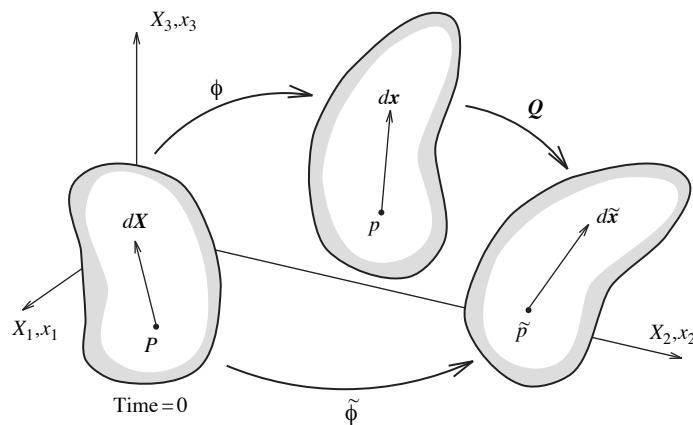
**Figure 2.18:** Lie derivative diagram.

$$\mathcal{L}_\phi[\mathbf{g}] = \phi_* \left[ \frac{d}{dt} (\phi_*^{-1}[\mathbf{g}]) \right]. \quad (2.118)$$

The Lie derivatives constitute the basic class of *objective* time derivatives of Eulerian objects. Lie derivatives are widely used in tensor analysis to distinguish between the change in some quantity and the change in that quantity but excluding the changes due to the motion or configuration changes; i.e. not depending on the frame of reference. For instance, an observer attached to the deforming material will see no changes between material and spatial configurations. However, the Eulerian observer will see changes to vectors and tensors. These changes are measured using the Lie derivative, which will be seen to be none other than the convective derivative term seen in Equation 2.105.

### 2.3.10. Objectivity

Objectivity is an important concept in solid mechanics. It is usually explored by studying the effect of a rigid body motion imposed on the deformed configuration as seen in Figure 2.19. For an observer attached to and rotating with the body, many quantities describing the behaviour of the solid will remain unchanged. For example, the distance between two particles or the stresses in the body, are said to be objective.



**Figure 2.19:** Superimposed rigid body motion [7].

Although the inherent nature of these quantities remains unchanged, their spatial description may change. To express these concepts an elemental vector  $d\mathbf{X}$  will be considered

in the initial configuration. This vector is deformed to  $d\mathbf{x}$ , and then is rotated to  $d\tilde{\mathbf{x}}$  as shown in Figure 2.19. The relationship between the vectors is given by

$$d\tilde{\mathbf{x}} = \mathbf{Q}d\mathbf{x} = \mathbf{Q}\mathbf{F} d\mathbf{X}, \quad (2.119)$$

where  $\mathbf{Q}$  is an orthogonal tensor describing the imposed rigid body rotation. Therefore, although the vector  $d\tilde{\mathbf{x}}$  is different from  $d\mathbf{x}$ , their magnitudes are equal. In this context, it is said that  $d\mathbf{x}$  is objective under rigid body motions.

Velocity is an example of a non-objective vector because when it is rotated,  $\tilde{\phi} = \mathbf{Q}\phi$ , the derivative with respect to time gives

$$\begin{aligned} \tilde{\mathbf{v}} &= \frac{\partial \tilde{\phi}}{\partial t} \\ &= \mathbf{Q} \frac{\partial \phi}{\partial t} + \dot{\mathbf{Q}}\phi \\ &= \mathbf{Q}\mathbf{v} + \dot{\mathbf{Q}}\phi. \end{aligned} \quad (2.120)$$

From the latter expression it is understood that the magnitudes  $\mathbf{v}$  and  $\tilde{\mathbf{v}}$  are not equal, and thus the magnitude is not objective. The definition of objectivity is extended to second-order tensors by introducing the relationship between the current and rotated configuration of the deformation gradient as

$$\tilde{\mathbf{F}} = \mathbf{Q}\mathbf{F}. \quad (2.121)$$

By introducing this expression into Equation 2.57 gives

$$\tilde{\mathbf{C}} = \tilde{\mathbf{F}}^T \tilde{\mathbf{F}} = \mathbf{F}^T \mathbf{Q}^T \mathbf{Q}\mathbf{F} = \mathbf{F}^T \mathbf{F} = \mathbf{C}, \quad (2.122)$$

and therefore, tensor  $\mathbf{C}$  and in turn  $\mathbf{E}$  are said to be objective since they remain unaltered by the rigid body motion. On the contrary, introducing Equation 2.119 into Equations 2.60 and 2.65 leads to

$$\tilde{\mathbf{b}} = \tilde{\mathbf{F}}\tilde{\mathbf{F}}^T = \mathbf{Q}\mathbf{F}\mathbf{F}^T\mathbf{Q}^T = \mathbf{Q}\mathbf{b}\mathbf{Q}^T; \quad (2.123)$$

$$\tilde{\mathbf{e}} = \mathbf{Q}\mathbf{e}\mathbf{Q}^T. \quad (2.124)$$

Although  $\tilde{\mathbf{e}} \neq \mathbf{e}$ , the definition of Almansi strain in Equation 2.64 shows that the spatial strain tensor is defined as the inner product and thus it is *pre* and *post* multiplied by the elemental vector  $d\mathbf{x} \cdot \mathbf{e} d\mathbf{x}$ . Introducing Equation 2.119 in the latter expression gives

$$d\mathbf{x} \cdot \mathbf{e} d\mathbf{x} = d\tilde{\mathbf{x}} \cdot \tilde{\mathbf{e}} d\tilde{\mathbf{x}}. \quad (2.125)$$

Here it is shown that although the strain quantities differ, the global change in the Almansi strain remains equal, and therefore it is objective.

## 2.4. Stress & Equilibrium

In the last sections the kinematics (or geometric relations) have been introduced, as well as the definitions of strain in both configurations. In this section the definition of stress is presented. Moreover, the set of differential equilibrium equations is derived, enforcing translational and rotational equilibrium, and the equivalent principle of virtual work is described.

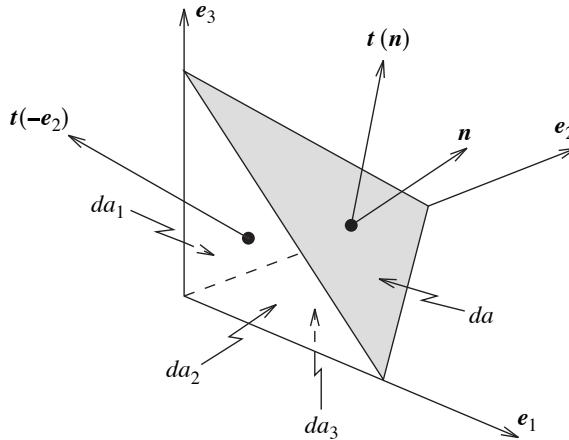
### 2.4.1. Cauchy Stress Tensor

The most fundamental and commonly used stress quantity is the *Cauchy stress*, also known as true stress. Both the force components and the area are related to the spatial configuration. To develop the stress tensor, first it is necessary to define the three traction vectors associated with the Cartesian directions  $e_1, e_2$  and  $e_3$  as

$$t(e_1) = \sigma_{11}e_1 + \sigma_{21}e_2 + \sigma_{31}e_3; \quad (2.126)$$

$$t(e_2) = \sigma_{12}e_1 + \sigma_{22}e_2 + \sigma_{32}e_3; \quad (2.127)$$

$$t(e_3) = \sigma_{13}e_1 + \sigma_{23}e_2 + \sigma_{33}e_3. \quad (2.128)$$



**Figure 2.20:** Elemental tetrahedron [8].

The relationship between the traction vector  $t$  corresponding to a general direction  $n$  and the components  $\sigma_{ij}$  can be obtained by studying the translational equilibrium of the elemental tetrahedron shown in Figure 2.20. Being  $f$  the force per unit volume acting on the body at point  $p$ , the equilibrium of the tetrahedron is

$$t(n) da + \sum_{i=1}^3 t(-e_i) da_i + f dv = 0, \quad (2.129)$$

where  $da_i = (n \cdot e_i) da$  is the projection of the area  $da$  onto the plane orthogonal to the Cartesian direction  $i$ , and  $dv$  is the volume of the tetrahedron. Dividing Equation 2.129 by  $da$ , using Equations 2.126, and noting that  $dv/da \rightarrow 0$  gives

$$\begin{aligned}
\mathbf{t}(\mathbf{n}) &= - \sum_{j=1}^3 \mathbf{t}(-\mathbf{e}_j) \frac{da_j}{da} - \mathbf{f} \frac{dv}{da} \\
&= \sum_{j=1}^3 \mathbf{t}(\mathbf{e}_j) (\mathbf{n} \cdot \mathbf{e}_j) \\
&= \sum_{i,j=1}^3 \sigma_{ij} (\mathbf{e}_j \cdot \mathbf{n}) \mathbf{e}_i \\
&= \sum_{i,j}^3 \sigma_{ij} (\mathbf{e}_i \otimes \mathbf{e}_j) \mathbf{n}.
\end{aligned} \tag{2.130}$$

The product  $(\mathbf{e}_j \cdot \mathbf{n}) \mathbf{e}_i$  can be rewritten in terms of the tensor product as  $(\mathbf{e}_i \otimes \mathbf{e}_j) \mathbf{n}$  giving

$$\mathbf{t}(\mathbf{n}) = \sum_{i,j=1}^3 \sigma_{ij} (\mathbf{e}_i \otimes \mathbf{e}_j) \mathbf{n}, \tag{2.131}$$

which identifies a tensor  $\boldsymbol{\sigma}$ , known as the Cauchy stress tensor, that relates the normal vector  $\mathbf{n}$  to the traction vector  $\mathbf{t}$  as

$$\mathbf{t}(\mathbf{n}) = \boldsymbol{\sigma} \mathbf{n}; \quad \boldsymbol{\sigma} = \sum_{i,j}^3 \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j. \tag{2.132}$$

As Cauchy stress tensor is a key quantity it will be necessary to see whether it is or not objective. Therefore, imposing a superimposed body motion  $\mathbf{Q}$ , as in Equation 2.121, the normal and tractions vectors are then

$$\tilde{\mathbf{t}}(\tilde{\mathbf{n}}) = \mathbf{Q} \mathbf{t}(\mathbf{n}); \tag{2.133}$$

$$\tilde{\mathbf{n}} = \mathbf{Q} \mathbf{n}. \tag{2.134}$$

Introducing the latter expressions into Equation 2.132 leads to

$$\tilde{\boldsymbol{\sigma}} = \mathbf{Q} \boldsymbol{\sigma} \mathbf{Q}^T. \tag{2.135}$$

Note that this expression has the same structure than Equations 2.123, and hence  $\boldsymbol{\sigma}$  is objective and a valid candidate for inclusion in a material description [8].

## 2.4.2. Equilibrium

### 2.4.2.1. Translational Equilibrium

To derive the differential static equilibrium equations we will consider a general deformable body defined by the characteristics of Table 2.4. Ignoring inertia forces, the translational equilibrium implies

Geometry		Forces	
Volume	Boundary area	Body forces	Traction forces
$v$	$\partial v$	$f$	$t$
[m <sup>3</sup> ]	[m <sup>2</sup> ]	[N/m <sup>3</sup> ]	[N/m <sup>2</sup> ]

**Table 2.4:** Characteristics of a general deformable body in 3D.

$$\int_{\partial v} \mathbf{t} da + \int_v \mathbf{f} dv = 0. \quad (2.136)$$

Using the Cauchy stress relation for the traction vector,

$$\int_{\partial v} \boldsymbol{\sigma} \mathbf{n} da + \int_v \mathbf{f} dv = 0. \quad (2.137)$$

Using the *Gauss theorem*<sup>8</sup>, the first term in the previous equation can be transformed into a volume integral to give

$$\int_v (\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}) dv = 0. \quad (2.138)$$

Note that this equation is valid for any enclosed region of the body, and thus the integrand vanishes

$$\mathbf{r} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = \mathbf{0}. \quad (2.139)$$

This equation is known as the *local spatial equilibrium equation* for a deformable body. In future situations during the iteration process, the residual  $\mathbf{r}$  may not be zero meaning that the equilibrium is not yet satisfied. The quantity  $\mathbf{r}$  is defined as the residual force per unit volume.

#### 2.4.2.2. Rotational Equilibrium

Although the Cauchy stress tensor is a symmetric tensor it has not been demonstrated yet. This is achieved by considering the rotational equilibrium of a general body under the action of traction and body forces. Therefore, the total moment of the body and traction forces about an arbitrary point must vanish as

$$\int_{dv} \mathbf{x} \times \mathbf{t} da + \int_v \mathbf{x} \times \mathbf{f} dv = \mathbf{0}. \quad (2.140)$$

Using the Cauchy stress tensor, this can be rewritten as

$$\int_{dv} \mathbf{x} \times (\boldsymbol{\sigma} \mathbf{n}) da + \int_v \mathbf{x} \times \mathbf{f} dv = \mathbf{0}. \quad (2.141)$$

<sup>8</sup>The Gauss' theorem is a result that relates the flux of a vector field through a surface to the behaviour of the vector field  $\mathbf{v}$  inside the surface as  $\int_V \nabla \cdot \mathbf{v} dV = \int_{\partial V} \mathbf{v} \cdot \mathbf{n} dA$ .

Using the Gauss theorem and after some algebra<sup>9</sup>, the equation becomes

$$\int_v \mathbf{x} \times (\nabla \cdot \boldsymbol{\sigma}) dv + \int_v \mathcal{E} : \boldsymbol{\sigma}^T dv + \int_v \mathbf{x} \times \mathbf{f} dv = \mathbf{0}, \quad (2.142)$$

where  $\mathcal{E}$  is the *Levi-Civita* third-order alternating tensor

$$\mathcal{E}_{ijk} \begin{cases} 0 & \text{if any of } i, j, k \text{ are the same} \\ +1 & \text{if } ijk \text{ is an even permutation} \\ -1 & \text{if } ijk \text{ is an odd permutation} \end{cases}$$

So, the vector  $\mathcal{E} : \boldsymbol{\sigma}^T$  is

$$\mathcal{E} : \boldsymbol{\sigma}^T = \begin{bmatrix} \sigma_{32} - \sigma_{23} \\ \sigma_{13} - \sigma_{31} \\ \sigma_{21} - \sigma_{12} \end{bmatrix}. \quad (2.143)$$

To take into account the translational equilibrium, the Equation 2.139 is substituted into Equation 2.142, and noting that the resulting equation is valid for any enclosed region of the body gives

$$\mathcal{E} : \boldsymbol{\sigma}^T = \mathbf{0}, \quad (2.144)$$

which, by observing Equation 2.143 implies the symmetry of the Cauchy stress tensor  $\boldsymbol{\sigma}$ .

### 2.4.3. Principle of Virtual Work

The principle of virtual work is discussed in Appendix A.1.1.2. However, it is introduced once again with tensor notation. In addition, the two different configurations (material and spatial) will be differentiated.

The virtual work  $\delta w$  per unit volume and time done by the residual force  $\mathbf{r}$  during a virtual motion is

$$\delta w = \mathbf{r} \cdot \delta \mathbf{v} = 0, \quad (2.145)$$

where  $\delta \mathbf{v}$  is the virtual velocity. We can introduce Equation 2.139 and integrate over the volume of the body to give a weak statement of the equilibrium as

$$\delta W = \int_v (\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}) \cdot \delta \mathbf{v} dv = 0, \quad (2.146)$$

using the following divergence property

$$\nabla \cdot (\boldsymbol{\sigma} \delta \mathbf{v}) = (\nabla \cdot \boldsymbol{\sigma}) \cdot \delta \mathbf{v} + \boldsymbol{\sigma} : \nabla \delta \mathbf{v}, \quad (2.147)$$

---

<sup>9</sup>The first addend in 2.141 is conveniently replaced by  $\int_{\delta v} \mathcal{E}_{ijk} x_j \sigma_{kl} n_l da$ . This expression is achieved using indicial notation and the summation convention. The Gauss theorem then gives the first two addends of Equation 2.142.

and the Gauss theorem we obtain

$$\int_{\delta v} \boldsymbol{\sigma} \cdot \delta \mathbf{v} \, da - \int_v \boldsymbol{\sigma} : \nabla \delta \mathbf{v} \, dv + \int_v \mathbf{f} \cdot \delta \mathbf{v} \, dv = 0, \quad (2.148)$$

recalling that

- the gradient of  $\delta \mathbf{v}$  is, by definition, the virtual velocity gradient  $\delta \mathbf{l}$ , and
- the Cauchy stress relation  $\boldsymbol{\sigma} \cdot \delta \mathbf{v} = \mathbf{t} \cdot \delta \mathbf{v}$ , we obtain

$$\int_{\delta v} \mathbf{t} \cdot \delta \mathbf{v} \, da - \int_v \boldsymbol{\sigma} : \delta \mathbf{l} + \int_v \mathbf{f} \cdot \delta \mathbf{v} \, dv = 0. \quad (2.149)$$

Finally, expressing the virtual velocity gradient in terms of the symmetric virtual rate of deformation  $\delta \mathbf{d}$  and taking into account the symmetry of  $\boldsymbol{\sigma}$ <sup>10</sup> the *spatial virtual work equation* can be written as

$$\delta W = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} \, dv - \int_v \mathbf{f} \cdot \delta \mathbf{v} \, dv - \int_{\delta v} \mathbf{t} \cdot \delta \mathbf{v} \, da. \quad (2.150)$$

This equation will become the basis for the finite element discretization.

#### 2.4.4. Work Conjugacy and Alternative Stress Representations

The principle of virtual work states that the internal work done by an infinitesimal strain variation operating on the current stresses equals the external work done by a corresponding virtual displacement operating on the loads. Due to the deformation of the body, the stress and strain measures must be selected so that their product gives an accurate energy density. This energy density may be related either to the material or spatial configuration, depending on whether the internal virtual work is integrated over the original or the deformed geometry.

##### 2.4.4.1. The Kirchhoff Stress Tensor

In Equation 2.149 the internal virtual work done by the stress is

$$\delta W_{int} = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} \, dv. \quad (2.151)$$

The quantities  $\boldsymbol{\sigma}$  and  $\mathbf{d}$  are said to be *work conjugate* with respect to the current volume in the sense that their product gives work per unit current volume. By expressing the virtual work equation in the material coordinate system, alternative work conjugate pairs emerge. As it is said previously, the reference configuration will be replaced with the deformed configuration and the limits of integration and variable will be changed. Thus, using Equation 2.80 to replace  $dv$  we obtain

---

<sup>10</sup>The deformation tensor  $\mathbf{d}$  expressed as the symmetric part of  $\mathbf{l}$  is  $\mathbf{d} = \frac{1}{2}(\mathbf{l} + \mathbf{l}^T)$ . Substituting this expression into  $\boldsymbol{\sigma} : \mathbf{d}$  gives  $\frac{1}{2}\boldsymbol{\sigma} : (\mathbf{l} + \mathbf{l}^T) = \frac{1}{2}\boldsymbol{\sigma}^T : \mathbf{l} + \frac{1}{2}\boldsymbol{\sigma} : \mathbf{l}^T = \boldsymbol{\sigma}^T : \mathbf{l}$ .

$$\int_V J\boldsymbol{\sigma} : \delta\mathbf{d} dV = \int_V \mathbf{f}_0 \cdot \delta\mathbf{v} dV + \int_{\partial V} \mathbf{t}_0 \cdot \delta\mathbf{v} dA, \quad (2.152)$$

where

- $\mathbf{f}_0 = J\mathbf{f}$  is the body force per unit undeformed volume, and
- $\mathbf{t}_0 = \mathbf{t}(da/dA)$  is the traction vector per unit initial area.

The internal virtual work given by the left-hand side of Equation 2.152 can be expressed in terms of the *Kirchhoff stress tensor*  $\boldsymbol{\tau}$  as

$$\delta W_{int} = \int_V \boldsymbol{\tau} : \delta\mathbf{d} dV; \quad \boldsymbol{\tau} = J\boldsymbol{\sigma}. \quad (2.153)$$

Note that the Kirchhoff stress tensor  $\boldsymbol{\tau}$  is work conjugate to the rate of deformation tensor with respect to the initial volume.

#### 2.4.4.2. The First Piola-Kirchhoff Stress Tensor

The previous transformation in the internal virtual work is not completely satisfactory because it still relies on the spatial quantities  $\boldsymbol{\tau}$  and  $\mathbf{d}$ . The symmetry of  $\boldsymbol{\sigma}$  together with Equation 2.109 for  $\mathbf{l}$  in terms of  $\dot{\mathbf{F}}$  and the properties of the trace<sup>11</sup> solves this lack of consistency to give

$$\begin{aligned} \delta W_{int} &= \int_V J\boldsymbol{\sigma} : \delta\mathbf{l} dV \\ &= \int_V J\boldsymbol{\sigma} : (\delta\dot{\mathbf{F}}\mathbf{F}^{-1}) dV \\ &= \int_V \text{tr}(J\mathbf{F}^{-1}\boldsymbol{\sigma}\delta\dot{\mathbf{F}}) dV \\ &= \int_V (J\boldsymbol{\sigma}\mathbf{F}^{-T}) : \delta\dot{\mathbf{F}} dV. \end{aligned} \quad (2.154)$$

We observe from this equality that the stress tensor work conjugate to the rate of the deformation gradient  $\dot{\mathbf{F}}$ . This new stress tensor is called the *first Piola-Kirchhoff stress tensor*, and it is defined as

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T}. \quad (2.155)$$

This first Piola-Kirchhoff stress tensor, like  $\mathbf{F}$ , is an unsymmetric two-point tensor with components given as

$$\mathbf{P} = \sum_{i,I=1}^3 P_{iI} \mathbf{e}_i \otimes \mathbf{E}_i; \quad P_{iI} = \sum_{j=1}^3 J\sigma_{ij} (\mathbf{F}^{-1})_{Ij}. \quad (2.156)$$

---

<sup>11</sup>The double product or double contraction of two tensors  $\mathbf{A}$  and  $\mathbf{B}$  is an invariant magnitude defined in terms of the trace as  $\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{A}^T \mathbf{B})$ .

We can now rewrite Equation 2.149 in terms of the Piola-Kirchhoff stress tensor as

$$\int_V \mathbf{P} : \delta \dot{\mathbf{F}} dV = \int_V \mathbf{f}_0 \cdot \delta \mathbf{v} dV + \int_{\delta V} \mathbf{t}_0 \cdot \delta \mathbf{v} dA. \quad (2.157)$$

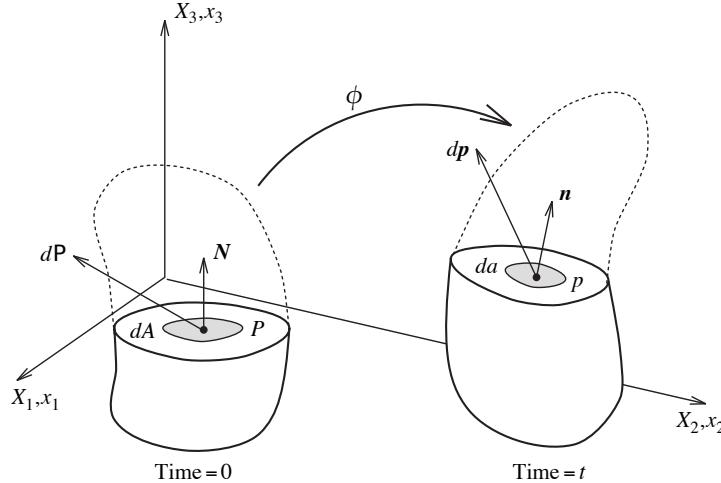
From Figure 2.21 it is deduced that an element of force  $d\mathbf{p}$  acting on an element of area  $d\mathbf{a} = \mathbf{n} da$  in the spatial configuration can be written as

$$d\mathbf{p} = \mathbf{t} da = \boldsymbol{\sigma} da. \quad (2.158)$$

So, the Cauchy stress gives the current force per unit deformed area, which is the classical definition of stress. Recalling expression 2.85 for the spatial area vector,  $d\mathbf{p}$  can be rewritten in terms of the undeformed area  $d\mathbf{A}$  to give an expression involving the first Piola-Kirchhoff stress as

$$d\mathbf{p} = J \boldsymbol{\sigma} \mathbf{F}^{-T} d\mathbf{A} = \mathbf{P} d\mathbf{A}. \quad (2.159)$$

The latter equation reveals that  $\mathbf{P}$ , as  $\mathbf{F}$ , is a two-point tensor that relates an area vector in the initial configuration to the force vector in the current configuration.



**Figure 2.21:** Interpretation of stress tensors [8].

#### 2.4.4.3. The Second Piola-Kirchhoff Stress Tensor

The first Piola-Kirchhoff tensor  $\mathbf{P}$  is unsymmetric and it is not completely related to the material configuration since it is defined using the current force element  $d\mathbf{p}$ . However, it is possible to obtain a symmetric material stress tensor, known as the *second Piola-Kirchhoff stress*  $\mathbf{S}$  by pulling back the force element  $d\mathbf{p}$  to give a material force vector  $d\mathbf{P}$  (shown in Figure 2.21) as

$$d\mathbf{P} = \phi_*^{-1}[d\mathbf{p}] = \mathbf{F}^{-1} d\mathbf{p}. \quad (2.160)$$

Introducing  $d\mathbf{p}$  from Equation 2.159 gives

$$d\mathcal{P} = \mathbf{S} dA; \quad \mathbf{S} = J\mathbf{F}^{-1}\boldsymbol{\sigma} \mathbf{F}^{-T}. \quad (2.161)$$

Now it is necessary to derive the strain rate work conjugate to the second Piola-Kirchhoff stress. Using the relation between the material and spatial rate of deformation Equation 2.115, the spatial virtual rate of deformation is

$$\delta\mathbf{d} = \mathbf{F}^{-T}\delta\dot{\mathbf{E}}\mathbf{F}^{-1}. \quad (2.162)$$

This virtual deformation  $\delta\mathbf{d}$  is now introduced into the internal virtual work Equation 2.154 to give

$$\begin{aligned} \delta W_{int} &= \int_v \boldsymbol{\sigma} : \delta\mathbf{d} dv \\ &= \int_V J\boldsymbol{\sigma} : (\mathbf{F}^{-T}\delta\dot{\mathbf{E}}\mathbf{F}^{-1}) dV \\ &= \int_V \text{tr}(\mathbf{F}^{-1}J\boldsymbol{\sigma}\mathbf{F}^{-T}\delta\dot{\mathbf{E}}) dV \\ &= \int_V \mathbf{S} : \delta\dot{\mathbf{E}} dV. \end{aligned} \quad (2.163)$$

which shows that  $\mathbf{S}$  is work conjugate to  $\dot{\mathbf{E}}$  and enables the *material virtual work* equation to be alternatively written in terms of  $\mathbf{S}$  as

$$\int_V \mathbf{S} : \delta\dot{\mathbf{E}} dV = \int_V \mathbf{f}_0 \cdot \delta\mathbf{v} dV + \int_{\delta V} \mathbf{t}_0 \cdot \delta\mathbf{v} dA. \quad (2.164)$$

The inverses of Equations 2.155 and 2.161 give

$$\boldsymbol{\sigma} = J^{-1}\mathbf{P}\mathbf{F}^T; \quad \boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^T. \quad (2.165)$$

Eventually, the second Piola-Kirchhoff and the Cauchy stresses are related by means of pull-back and push-forward operations as

$$\mathbf{S} = J\phi_*^{-1}[\boldsymbol{\sigma}]; \quad \boldsymbol{\sigma} = J^{-1}\phi_*[\mathbf{S}]. \quad (2.166)$$

All the equilibrium equations obtained in this section are written in Table 2.5.

#### 2.4.5. Physical Interpretation of Stresses

All the stresses explained as well as the related strain and deformation rates are summarized in Table 2.6. In this section it is intended to describe slightly the physical meaning of all the stress quantities. The most common stress quantity is the Cauchy or true stress  $\boldsymbol{\sigma}$ . It is defined by the current force and the deformed body. Both the force and the normal to the area have fixed directions in space. This means that if a stressed body is subjected to pure rotation, the value of the stress will change. Sometimes this feature can be undesirable.

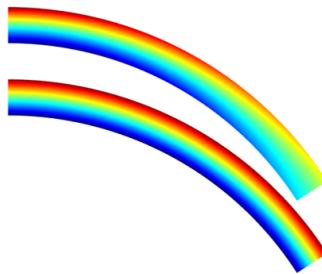
Configuration	Equilibrium equations
<i>Reference</i>	$\int_v \mathbf{S} : \delta \dot{\mathbf{E}} dV = \int_v \mathbf{f}_0 \cdot \delta \mathbf{v} dv + \int_{\partial v} \mathbf{t}_0 \cdot \delta \mathbf{v} da$
<i>In-between</i>	$\int_V \mathbf{P} : \delta \dot{\mathbf{F}} dV = \int_V \mathbf{f}_0 \cdot \delta \mathbf{v} dV + \int_{\partial V} \mathbf{t}_0 \cdot \delta \mathbf{v} dA$
<i>Updated</i>	$\int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv = \int_v \mathbf{f} \cdot \delta \mathbf{v} dv + \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} da$

**Table 2.5:** Equilibrium equations.

The second Piola-Kirchhoff stress  $\mathbf{S}$  can solve this problem. In the case of rigid body rotation the components of  $\mathbf{S}$  coincide with the components of  $\boldsymbol{\sigma}$  expressed in the local set of orthogonal axes. These axes result from rotating the global Cartesian directions according the rotation tensor  $\mathbf{R}$ . This rotation, recalling the polar decomposition, is expressed as  $\mathbf{F} = \mathbf{R}$  and  $J = 1$ , and thus

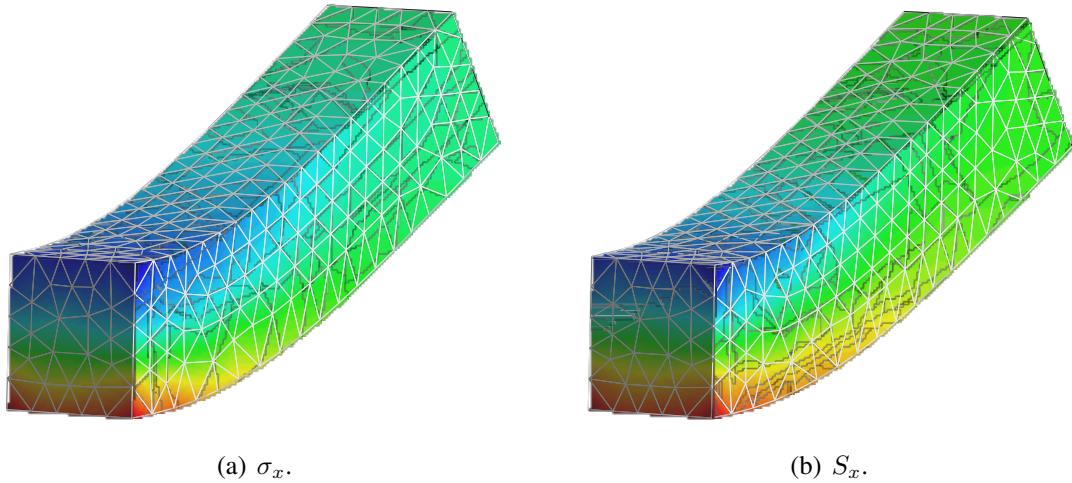
$$\mathbf{S} = \mathbf{R}^T \boldsymbol{\sigma} \mathbf{R}. \quad (2.167)$$

This last expression corresponds to a common coordinate transformation. This can be interesting, for example in the case of using an orthotropic material with fibers. It would be more logical visualizing the stress in the fiber direction, even if the component is rotated [9]. The second Piola-Kirchhoff stress has this property.

**Figure 2.22:** Cauchy (top) and second Piola-Kirchhoff stress (bottom) for beam with constant bending moment [9].

From Figure 2.22 shows a cantilever beam subjected to bending by a pure moment at the tip. The stresses plotted are  $\sigma_{xx}$  and  $S_{xx}$ . Since the stress is physically directed along the beam, the  $xx$  component of the Cauchy stress, which is related to the global  $x$  axis, decreases with the deflection. However, the second Piola-Kirchhoff stress maintains the thickness through the beam, even in the deformed configuration. Figure 2.23 shows the distribution of both stresses for a beam structure. This result is obtained in Chapter 4, and from it can be observed the trend of second Piola-Kirchhoff stress to maintain the path along the structure.

The first Piola-Kirchhoff stress is a multiaxial generalization of the engineering stress. It is defined as the force in the current configuration acting on the original area. The first Piola-Kirchhoff tensor is unsymmetric and thus it is less attractive to work with.



**Figure 2.23:** Cauchy and second Piola-Kirchhoff stress distributions.

Eventually, we find the Kirchhoff stress. This tensor is just the Cauchy stress scaled by the volume change [9]. It has little physical significance, however it is useful in some mathematical operations.

Configuration				
	Reference	In-between	Updated	
Strain	$\dot{\mathbf{E}}$	$\dot{\mathbf{F}}$	$\mathbf{d}$	$\mathbf{d}$
Stress	$\mathbf{S}$	$\mathbf{P}$	$\boldsymbol{\tau}$	$\boldsymbol{\sigma}$
Symmetry	✓	✗	✓	✓
Volume	$dV$	$dV$	$dV$	$dv$

**Table 2.6:** Résumé of all conjugate stress-strain tensor pairs.

## 2.5. Constitutive Model

The equilibrium equations exhibited in Table 2.5 are written in terms of the stresses inside the body. These stresses result from the deformation of the material, being the strain the most common way to measure it. The relationship between stress and strain is known as constitutive equation, and depend on the microstructure of the material - what constitutes that material. Together with kinematic and equilibrium equations, the constitutive equations allows the mechanics problem to be solved.

The constitutive equation will be established in the context of a hyperelastic material given its simplicity. Moreover, hyperelasticity constitutes the basis for more complex material models such as elastoplasticity and viscoplasticity [10].

For an hyperelastic material the stresses defined in the constitutive equation arise from a stored elastic energy function  $\Psi$ .

### 2.5.1. Hyperelasticity

Hyperelasticity provides higher-order forms of lineal elasticity, and constitutes one of the nonlinear elastic models. Therefore, hyperelastic models are used to model materials which respond elastically when subjected to very large strains. They account both for nonlinear material behaviour and large geometric changes.

In elastic materials, the stress is directly proportional to strain by using a linear relationship A.63. In these materials the constitutive behaviour is only a function of the current state of deformation. Any stress measure at a particle  $\mathbf{X}$  is a function of the current deformation gradient  $\mathbf{F}$  associated with that particle. Instead of using any of the alternative strain measures, the deformation gradient  $\mathbf{F}$ , together with its conjugate first Piola-Kirchhoff stress measure  $\mathbf{P}$ . Consequently, assuming the homogeneity of the material, elasticity can be expressed as

$$\mathbf{P} = \mathbf{P}(\mathbf{F}(\mathbf{X})). \quad (2.168)$$

Nevertheless, elastic solids subjected to large deformations will not follow linear elasticity. When the work done by the stresses during a deformation process is dependent only on the initial state at time  $t_0$  and the final configuration at time  $t$ , the material is said to be path-independent and the material is called hyperelastic.

A stored strain energy function or elastic potential  $\Psi$  per unit undeformed volume can be established as the work done by the stresses from the initial to the current position as

$$\Psi(\mathbf{F}(\mathbf{X})) = \int_{t_0}^t \mathbf{P}(\mathbf{F}(\mathbf{X})) : \dot{\mathbf{F}} dt. \quad (2.169)$$

The rate of change of the potential (2.169) can be alternatively expressed as

$$\dot{\Psi} = \mathbf{P} : \dot{\mathbf{F}} = \frac{\partial \Psi(\mathbf{F}(\mathbf{X}))}{\partial \mathbf{F}} \dot{\mathbf{F}}. \quad (2.170)$$

From physical experiments it is possible to construct the function  $\Psi(\mathbf{F})$  which defines a type of material. This function must obey certain conditions, such as objectivity. The objectivity principle states that  $\Psi$  must remain invariant when the current configuration undergoes a rigid body rotation. This implies that  $\Psi$  depends on  $\mathbf{F}$  only via the stretch component  $\mathbf{U}$  and it is independent of the rotation component  $\mathbf{R}$ . However, for convenience,  $\Psi$  is expressed as a function of  $\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^T \mathbf{F}$  as

$$\Psi(\mathbf{F}(\mathbf{X})) = \Psi(\mathbf{C}(\mathbf{X})). \quad (2.171)$$

Thus, 2.170 can be expressed as

$$\dot{\Psi} = \frac{\partial \Psi}{\partial \mathbf{C}} : \dot{\mathbf{C}}. \quad (2.172)$$

Observing that  $\dot{\mathbf{E}}$  is work conjugate to the second Piola-Kirchhoff stress  $\mathbf{S}$ , it is possible to obtain a totally lagrangian constitutive equation to give

$$\dot{\Psi} = \mathbf{S} : \dot{\mathbf{E}}. \quad (2.173)$$

Eventually, the relation  $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$  enables a totally lagrangian constitutive equation, where the second Piola-Kirchhoff stress  $\mathbf{S}$  is computed from the elastic potential  $\Psi$ , defined as

$$\dot{\Psi} = \frac{1}{2}\mathbf{S} : \dot{\mathbf{C}}, \quad (2.174)$$

where  $\mathbf{S}$  can be defined as,

$$\mathbf{S}(\mathbf{C}(\mathbf{X})) = 2\frac{\partial\Psi}{\partial\mathbf{C}} = \frac{\partial\Psi}{\partial\mathbf{E}}. \quad (2.175)$$

### 2.5.2. Elasticity Tensor

Owing to the essential nonlinearity of the hyperelastic model, the relationship between  $\mathbf{S}$  and  $\mathbf{C}$  or  $\mathbf{E}$  in 2.175 will need to be linearized with respect to an increment  $\mathbf{u}$  in the current configuration as

$$D\mathbf{S}[\mathbf{u}] = 4\frac{\partial^2\Psi}{\partial\mathbf{C}\partial\mathbf{C}} : D\mathbf{E}[\mathbf{u}]. \quad (2.176)$$

The first term of 2.176 is a symmetric fourth-order tensor  $\mathcal{C}$  named Lagrangian or material elasticity tensor, and it is often expressed as

$$\mathcal{C} = 2\frac{\partial\mathbf{S}}{\partial\mathbf{C}} = 4\frac{\partial^2\Psi}{\partial\mathbf{C}\partial\mathbf{C}}. \quad (2.177)$$

The Lagrangian elasticity tensor is defined by the partial derivatives of strain and stress as

$$\mathcal{C} = \sum_{I,J,K,L=1}^3 \mathcal{C}_{IJKL} \mathbf{E}_I \otimes \mathbf{E}_J \otimes \mathbf{E}_K \otimes \mathbf{E}_L; \quad \mathcal{C}_{IJKL} = \frac{\partial S_{IJ}}{\partial E_{KL}} = \frac{4\partial^2\Psi}{\partial C_{IJ}\partial C_{KL}}. \quad (2.178)$$

Note the similarity between  $\mathcal{C}$  and the constitutive elasticity tensor. However, the Lagrangian elasticity tensor is not a constant, but a function of strain  $\mathcal{C}(\mathbf{C})$ . Since the stress and strain variables belong to the material configuration a push-forward operation will be required to compute the Eulerian or spatial elasticity tensor  $\mathbf{c}$ ,

$$\mathbf{c} = J^{-1}\phi_*[\mathcal{C}]. \quad (2.179)$$

The spatial elasticity tensor expressed in index notation is written as

$$c = \sum_{\substack{i,j,k,l=1 \\ I,J,K,L=1}}^3 J^{-1} F_{iI} F_{jJ} F_{kK} F_{lL} C_{IJKL} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l. \quad (2.180)$$

### 2.5.3. Isotropic Hyperelasticity

In order to restrict the constitutive equations, it is often introduced the isotropic material. An isotropic material exhibits the same identical behaviour in any material direction. Therefore, the relationship between  $\Psi$  and  $C$  is independent of the material axes chosen, being  $\Psi$  only function of the invariants<sup>12</sup> of  $C$  as

$$\Psi(C(\mathbf{X})) = \Psi(I_1, I_2, I_3), \quad (2.181)$$

where the invariants of  $C$  are defined as

$$I_1 = \text{tr } C = C : \mathbf{I}; \quad (2.182)$$

$$I_2 = \text{tr } CC = C : C; \quad (2.183)$$

$$I_3 = \det C = J^2. \quad (2.184)$$

As a result of the isotropic restriction, the second Piola-Kirchhoff stress tensor can be written as

$$\mathbf{S} = 2 \frac{\partial \Psi}{\partial C} = 2 \frac{\partial \Psi}{\partial I_1} \frac{\partial I_1}{\partial C} + 2 \frac{\partial \Psi}{\partial I_2} \frac{\partial I_2}{\partial C} + 2 \frac{\partial \Psi}{\partial I_3} \frac{\partial I_3}{\partial C}. \quad (2.185)$$

The second-order tensors formed by the derivatives of the invariants result in,

$$\frac{\partial I_1}{\partial C} = \mathbf{I}; \quad \frac{\partial I_2}{\partial C} = 2C; \quad \frac{\partial I_3}{\partial C} = J^2 C^{-1}. \quad (2.186)$$

Introducing Expressions 2.186 into Equation 2.185, the second Piola-Kirchhoff stress can be evaluated as

$$\mathbf{S} = 2\Psi_1 \mathbf{I} + 4\Psi_2 C + 2J^2 \Psi_3 C^{-1}, \quad (2.187)$$

where  $\Psi_1 = \partial \Psi / \partial I_1$ ,  $\Psi_2 = \partial \Psi / \partial I_2$ ,  $\Psi_3 = \partial \Psi / \partial I_3$ .

The Cauchy stresses can be obtained indirectly from the second Piola-Kirchhoff stresses by using

$$\boldsymbol{\sigma} = J^{-1} \mathbf{F} \mathbf{S} \mathbf{F}^T. \quad (2.188)$$

---

<sup>12</sup>Invariants of tensors are coefficients of the characteristic polynomial of the tensor.

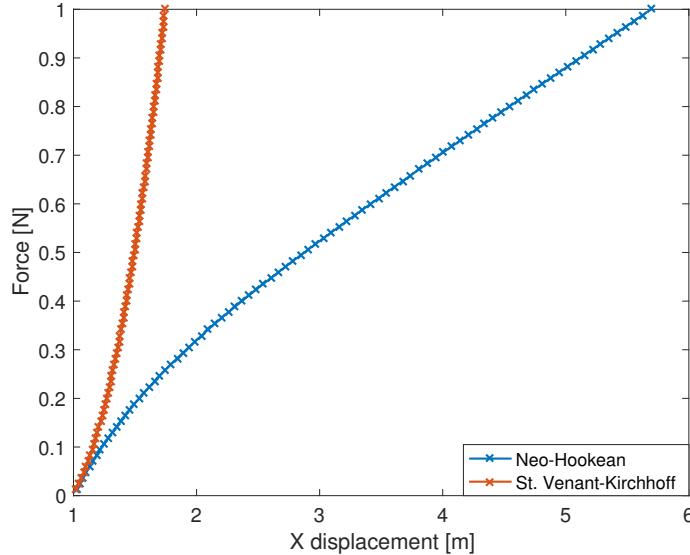
In fact, the Cauchy stress is the push-forward of the second Piola-Kirchhoff stress. Substituting  $\mathbf{S}$  from Equation 2.187, and considering that the left Cauchy-Green tensor is  $\mathbf{b} = \mathbf{F}\mathbf{F}^T$  gives

$$\boldsymbol{\sigma} = 2J^{-1}\Psi_1\mathbf{b} + 4J^{-1}\Psi_2\mathbf{b}^2 + 2J\Psi_3\mathbf{I}. \quad (2.189)$$

In the aforementioned Equation,  $\Psi_1$ ,  $\Psi_2$  and  $\Psi_3$  also involve derivatives with respect to the invariants of the material tensor  $\mathbf{C}$ . However, these invariants are identical to the invariants of  $\mathbf{b}$  [10].

#### 2.5.4. Constitutive Material Models

The linear elastic material model, as described in Section A.2.2, is only valid for small strains and small rotations. The modification to this model to the range of nonlinear deformations will be given by the Saint Venant-Kirchhoff and Neo-Hookean models. These compressible models only accounts for geometric nonlinearity, however, material nonlinearity can be neglected for deflections smaller than  $20h$ , being  $h$  the thickness of the beam [11]. Nevertheless, as it is explained in Section 3.2, the use of pull-push operations introduces a generalization in the code allowing different implementations of other hyperelastic models.



**Figure 2.24:** Force versus displacement for Saint Venant-Kirchhoff and Neo-Hookean models. This response diagram corresponds to the *test* result (Section 4.1).

The stored energy expressions consider some parameters which arise from the behaviour of the material, such as the Young's modulus  $E$  and the Poisson coefficient  $\nu$ . The Lamé constants are derived from the latter parameters as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}; \quad \mu = \frac{E}{2(1+\nu)}. \quad (2.190)$$

The Young's modulus is a measure of the stiffness of the material, while the Poisson's ratio is a measure of the incompressibility of the material and it ranges from 0 to  $\frac{1}{2}$ .

#### 2.5.4.1. Saint Venant-Kirchhoff

A wide class of engineering problems can be studied by linear elastic material behaviour. If the effects of large deformation are primarily due to geometric nonlinearities (such as in the bending of a wing) a straightforward generalization of Hooke's law to finite strains is often adequate. The Saint Venant-Kirchhoff model accomplishes this through the use of the Green strain measure  $\mathbf{E}$ .

The St. Venant-Kirchhoff model, is defined by the following strain energy function,

$$\Psi = \frac{1}{2}\lambda(\text{tr } \mathbf{E})^2 + \mu\mathbf{E} : \mathbf{E}. \quad (2.191)$$

By using the second part of Equation 2.175, the second Piola-Kirchhoff stress tensor is defined as

$$\mathbf{S} = \frac{\partial\Psi(\mathbf{E})}{\partial\mathbf{E}} = \mathcal{C} : \mathbf{E}. \quad (2.192)$$

The Lagrangian elasticity tensor is analogous to the linear elasticity case. Then, the Lagrangian elasticity tensor coefficients are

$$\mathcal{C}_{IJKL} = \lambda\delta_{IJ}\delta_{KL} + \mu(\delta_{IK}\delta_{JL} + \delta_{IL}\delta_{JK}), \quad (2.193)$$

and then, the second Piola-Kirchhoff stress is

$$\mathbf{S} = \lambda(\text{tr } \mathbf{E})\mathbf{I} + 2\mu\mathbf{E}. \quad (2.194)$$

To compute the Cauchy stress tensor Equation 2.188 is used. The Saint Venant-Kirchhoff model possesses well-known limitations, particularly some instabilities when subjected to pure compression [12]. Moreover, it has little practical use beyond the small strain regime. Nevertheless, the St. Venant-Kirchhoff material is appropriate for modelling metals and ceramics, which fracture or yield at small deformations.

#### 2.5.4.2. Compressible Neo-Hookean

The large strain Hookean or neo-Hookean model appeared during World War II as rubber came into use as an engineering material, and a need arose to express Hooke's law for large deformations [13].

The stored energy function for a compressible Neo-Hookean material is written as

$$\Psi = \frac{\mu}{2}(I_1 - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2. \quad (2.195)$$

It must be observed that in the absence of deformation, the right Cauchy-Green deformation tensor  $\mathbf{C} = \mathbf{I}$ , and thus the stored energy function vanishes. From Equations 2.187 and 2.189, the second Piola-Kirchhoff and Cauchy stress tensors are respectively

$$\mathbf{S} = \mu(\mathbf{I} - \mathbf{C}^{-1}) + \lambda(\ln J)\mathbf{C}^{-1}; \quad (2.196)$$

$$\boldsymbol{\sigma} = \frac{\mu}{J}(\mathbf{b} - \mathbf{I}) + \frac{\lambda}{J}(\ln J)\mathbf{I}. \quad (2.197)$$

The elasticity tensor is obtained by differentiating  $\mathbf{S}$  using Equation 2.187, and after some algebra  $\mathbf{C}$  is expressed as

$$\mathbf{C} = \lambda\mathbf{C}^{-1} \otimes \mathbf{C}^{-1} + 2(\mu - \lambda \ln J)\mathbf{I}, \quad (2.198)$$

where  $\mathbf{C}^{-1} \otimes \mathbf{C}^{-1} = (\mathbf{C}^{-1})_{IJ}(\mathbf{C}^{-1})_{KL}\mathbf{E}_I \otimes \mathbf{E}_J \otimes \mathbf{E}_K \otimes \mathbf{E}_L$ , and the components of the fourth-order tensor  $\mathbf{I}$  is defined as

$$\mathcal{I}_{IJKL} = \frac{1}{2}[(\mathbf{C}^{-1})_{IK}(\mathbf{C}^{-1})_{JL} + (\mathbf{C}^{-1})_{IL} + (\mathbf{C}^{-1})_{JK}]. \quad (2.199)$$

## 2.6. Linearized Equilibrium Equations

The process of linearization provides a key link between the linear and nonlinear theories of elasticity. It is one of the most crucial aspects of nonlinear analysis, since the virtual work representation of equilibrium equations is nonlinear with respect to both the geometry and the material (in our case there is only geometrical nonlinearity).

### 2.6.1. Linearization and Newton-Raphson Process

The principle of virtual work has been expressed in different configurations in Section 2.4. Taking the spatial equilibrium equation, in terms of the virtual velocity as

$$\delta W(\phi, \delta \mathbf{v}) = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv - \int_v \mathbf{f} \cdot \delta \mathbf{v} dv - \int_{\delta v} \mathbf{t} \cdot \delta \mathbf{v} da = 0. \quad (2.200)$$

Now, a trial solution  $\phi_k$  is considered. Equation 2.200 can be linearized in the direction of an increment  $\mathbf{u}$  in  $\phi_k$  as<sup>13</sup>

$$\delta W(\phi, \delta \mathbf{v}) = \delta W(\phi_k, \delta \mathbf{v}) + D\delta W(\phi_k, \delta \mathbf{v})[\mathbf{u}] = 0. \quad (2.201)$$

It is necessary to comment three important aspects arisen from the latter expression.

- It is necessary to find the directional derivative of the virtual work equation at  $\phi_k$  in the direction of  $\mathbf{u}$ . The virtual velocity  $\delta \mathbf{v}(\phi(\mathbf{X}))$ , associated with each  $\mathbf{X}$  particle in the body, is not allowed to alter during the incremental change  $\mathbf{u}(\mathbf{x})$ .

---

<sup>13</sup>The symbol  $Df(x_0)[u]$  denotes a derivative, formed at  $x_0$ , that operates in some linear manner on  $u$  (not necessarily a multiplication).

- The directional derivative  $D\delta W(\phi_k, \delta \mathbf{v})$  is simply the change in  $\delta W$  due to  $\phi_k$  changing to  $\phi_k + \mathbf{u}$ . Since  $\delta \mathbf{v}$  remains constant during this change, the directional derivative must represent the change in the internal forces due to  $\mathbf{u}$  (assuming that external forces are constant).
- Hence, the directional derivative of the virtual work equation will be the source of the tangent matrix  $\mathbf{K}$  (the slope of the stress-strain curve at any specified point).

To linearize the equilibrium equation it will be separated in internal and external virtual work components as

$$D\delta W(\phi, \delta \mathbf{v})[\mathbf{u}] = D\delta W_{int}(\phi, \delta \mathbf{v})[\mathbf{u}] - D\delta W_{ext}(\phi, \delta \mathbf{v})[\mathbf{u}], \quad (2.202)$$

where,

$$\delta W_{int}(\phi, \delta \mathbf{v}) = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv; \quad (2.203)$$

$$\delta W_{ext}(\phi, \delta \mathbf{v}) = \int_v \mathbf{f} \cdot \delta \mathbf{v} dv + \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} da. \quad (2.204)$$

### 2.6.2. Lagrangian Linearized Internal Virtual Work

The eventual discretization of the linearized equilibrium equations is formulated only for the Eulerian case. However, it is convenient to perform the linearization with respect to the material description because the initial elemental volume  $dV$  is constant. Therefore, by using the push-forward operation, the linearized equations in terms of the spatial configuration are obtained. Recalling internal virtual work can be expressed using Equation 2.163 as

$$\delta W_{int}(\phi, \delta \mathbf{v}) = \int_V \mathbf{S} : \delta \dot{\mathbf{E}} dV. \quad (2.205)$$

Using the product rule for directional derivatives and the definition of the material elasticity tensor  $\mathcal{C}$ , the directional derivative is obtained as

$$\begin{aligned} D\delta W_{int}(\phi, \delta \mathbf{v}) &= \int_V D(\delta \dot{\mathbf{E}} : \mathbf{S})[\mathbf{u}] dV \\ &= \int_V \delta \dot{\mathbf{E}} : D\mathbf{S}[\mathbf{u}] dV + \int_V \mathbf{S} : D\delta \dot{\mathbf{E}}[\mathbf{u}] dV \\ &= \int_V \delta \dot{\mathbf{E}} : \mathcal{C} : D\mathbf{E}[\mathbf{u}] dV + \int_V \mathbf{S} : D\delta \dot{\mathbf{E}}[\mathbf{u}] dV. \end{aligned} \quad (2.206)$$

This computation is achieved by using  $D\mathbf{S}[\mathbf{u}] = \mathcal{C} : D\mathbf{E}[\mathbf{u}]$  (Equation 2.176), where  $D\mathbf{E}[\mathbf{u}]$  is given by

$$D\mathbf{E}[\mathbf{u}] = \frac{1}{2}\mathbf{F}^T[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]\mathbf{F}. \quad (2.207)$$

The term  $D\delta\dot{\mathbf{E}}[\mathbf{u}]$  in the second integral (expression 2.206) emerges from the fact that  $\delta\dot{\mathbf{E}}$  is a function not only of  $\delta\mathbf{v}$  but also of the configuration  $\phi$  as

$$\delta\dot{\mathbf{E}} = \frac{1}{2}(\delta\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\delta\dot{\mathbf{F}}); \quad \delta\dot{\mathbf{F}} = \frac{\partial\delta\mathbf{v}}{\partial\mathbf{X}} = \nabla_0\delta\mathbf{v}. \quad (2.208)$$

The direction derivative of the latter equation can be found recalling from

- $D\mathbf{F}[\mathbf{u}] = \nabla_0\mathbf{u};$
- $D\mathbf{F}[\mathbf{v}] = \nabla_0\mathbf{v} = \dot{\mathbf{F}},$

to give

$$D\delta\dot{\mathbf{E}}[\mathbf{u}] = \frac{1}{2}[(\nabla_0\delta\mathbf{v})^T\nabla_0\mathbf{u} + (\nabla_0\mathbf{u})^T\nabla_0\delta\mathbf{v}]. \quad (2.209)$$

Observing that virtual velocities  $\delta\mathbf{v}$  are not a function of the configuration and thus  $\nabla_0\delta\mathbf{v}$  remains constant, and being  $\mathbf{S}$  symmetric and  $\mathbf{C}$  a general tensor;  $\mathbf{S}\frac{1}{2}(C^T + C) = \frac{1}{2}\mathbf{SC}^T + \frac{1}{2}\mathbf{SC}$ , and thus  $\mathbf{SC}^T = \mathbf{SC}$ , the material linearized principle of virtual work is

$$D\delta W_{int}(\phi, \delta\mathbf{v})[\mathbf{u}] = \int_V \delta\dot{\mathbf{E}} : \mathbf{C} : D\mathbf{E}[\mathbf{u}] dV + \int_V \mathbf{S} : [(\nabla_0\mathbf{u})^T\nabla_0\delta\mathbf{v}] dV. \quad (2.210)$$

The relation between directional and time derivatives  $\delta\dot{\mathbf{E}} = D\mathbf{E}[\delta\mathbf{v}]$ , enables to write the equation in a more obviously symmetric form,

$$D\delta W_{int}(\phi, \delta\mathbf{v})[\mathbf{u}] = \int_V D\mathbf{E}[\delta\mathbf{v}] : \mathbf{C} : D\mathbf{E}[\mathbf{u}] dV + \int_V \mathbf{S} : [(\nabla_0\mathbf{u})^T\nabla_0\delta\mathbf{v}] dV. \quad (2.211)$$

### 2.6.3. Eulerian Linearized Internal Virtual Work

Equation 2.211 could be used for the development of the tangent stiffness matrix. Nevertheless, much simplification is gained by employing the equivalent spatial alternative to give the same tangent matrix [14]. Therefore, applying the pull-back operations to the materially based terms in Equation 2.211 as

$$D\mathbf{E}[\mathbf{u}] = \phi_*^{-1}[\boldsymbol{\varepsilon}] = \mathbf{F}^T\boldsymbol{\varepsilon}\mathbf{F}; \quad 2\boldsymbol{\varepsilon} = \nabla\mathbf{u} + (\nabla\mathbf{u})^T; \quad (2.212)$$

$$D\mathbf{E}[\delta\mathbf{v}] = \phi_*^{-1}[\delta\mathbf{d}] = \mathbf{F}^T\delta\mathbf{d}\mathbf{F}; \quad 2\delta\mathbf{d} = \nabla\delta\mathbf{v} + (\nabla\delta\mathbf{v})^T, \quad (2.213)$$

and the corresponding push-forward operations

$$J\boldsymbol{\sigma} = \phi_*[\boldsymbol{S}] = \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T; \quad (2.214)$$

$$J\boldsymbol{c} = \phi_*[\boldsymbol{C}]; \quad (2.215)$$

$$J dV = dv. \quad (2.216)$$

The first part of the integrand of Equation 2.211 can be re-expressed in a spatial framework as

$$D\boldsymbol{E}[\delta\boldsymbol{v}]:\boldsymbol{C}:D\boldsymbol{E}[\boldsymbol{u}] dV = \delta\boldsymbol{d}:\boldsymbol{c}:\boldsymbol{\varepsilon} dv. \quad (2.217)$$

The gradient with respect to the initial particle coordinates appearing in the second integral of 2.211 can be related to the spatial gradient using the chain rule as

$$\nabla_0\boldsymbol{u} = (\nabla\boldsymbol{u})\boldsymbol{F}; \quad (2.218)$$

$$\nabla_0\delta\boldsymbol{v} = (\nabla\delta\boldsymbol{v})\boldsymbol{F}. \quad (2.219)$$

Substituting these expressions into the second term of Equation 2.211, and using the expression 2.214 for the Cauchy stress gives

$$\boldsymbol{S}:[(\nabla_0\boldsymbol{u})^T(\nabla_0\delta\boldsymbol{v})] dV = \boldsymbol{\sigma}:[(\nabla\boldsymbol{u})^T(\nabla\delta\boldsymbol{v})] dv. \quad (2.220)$$

The functional relationship between  $\delta\boldsymbol{d}$  and  $\delta\boldsymbol{v}$  is identical to  $\boldsymbol{\varepsilon}$  and  $\boldsymbol{u}$  (see Equation 2.212). This together with the symmetry of  $\boldsymbol{c}$  and  $\boldsymbol{\sigma}$  implies that the terms  $\boldsymbol{u}$  and  $\delta\boldsymbol{v}$  can be interchanged in this equation without altering the result. This is expressed as

$$D\delta W_{int}(\phi, \delta\boldsymbol{v})[\boldsymbol{u}] = D\delta W_{int}(\phi, \boldsymbol{u})[\delta\boldsymbol{v}]. \quad (2.221)$$

The linearized virtual work equation is symmetric in  $\delta\boldsymbol{v}$  and  $\boldsymbol{u}$ . Thus, this symmetry yield a symmetric tangent stiffness matrix. For future sections it is convenient to divide the internal virtual work into constitutive and initial stress components as

$$\begin{aligned} D\delta W_{int}(\phi, \delta\boldsymbol{v})[\boldsymbol{u}] &= D\delta W_c(\phi, \delta\boldsymbol{v})[\boldsymbol{u}] + D\delta W_\sigma(\phi, \delta\boldsymbol{v})[\boldsymbol{u}] \\ &= \int_v \delta\boldsymbol{d}:\boldsymbol{c}:\boldsymbol{\varepsilon} dv + \int_v \boldsymbol{\sigma}:[(\nabla\boldsymbol{u})^T\nabla\delta\boldsymbol{v}] dv. \end{aligned} \quad (2.222)$$

#### 2.6.4. Linearized External Virtual Work

The external virtual work has contributions from body forces  $\boldsymbol{f}$  and surface tractions  $\boldsymbol{t}$ .

### 2.6.4.1. Body Forces

The most common example is self-weight or gravity loading in which case  $\mathbf{f} = \rho\mathbf{g}$ . By a simple pull-back of body forces it is easy to show that in this case the loading is not deformation-dependent, and therefore the corresponding directional derivative vanishes.

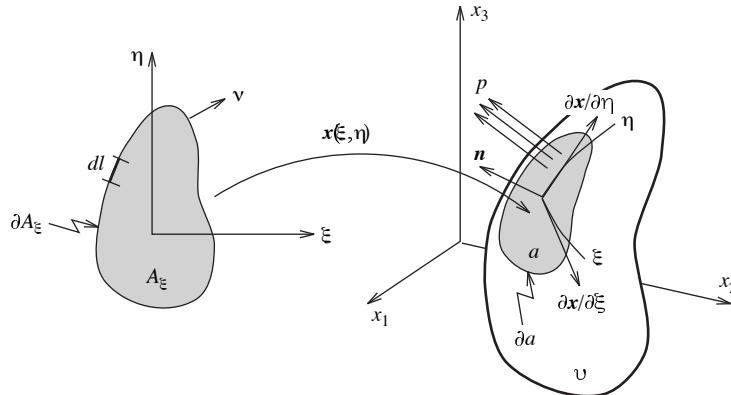
$$\delta W_{ext}^f(\phi, \delta\mathbf{v}) = \int_v \frac{\rho_0}{J} \mathbf{g} \cdot \delta\mathbf{v} dv = \int_V \rho_0 \mathbf{g} \cdot \delta\mathbf{v} dV. \quad (2.223)$$

It is clear that none of the terms in this expression depend on the current geometry, and hence its linearization is superfluous.

$$D\delta W_{ext}^f(\phi, \delta\mathbf{v})[\mathbf{u}] = 0. \quad (2.224)$$

### 2.6.4.2. Surface Forces

In here, only the important case of uniform normal pressure will be addressed. However, a similar technique can be used in more complex situations such as frictional contact.



**Figure 2.25:** Uniform surface pressure [14].

Figure 2.25 shows a general body with an applied uniform pressure  $p$  acting on a surface  $a$ . The traction force vector  $\mathbf{t}$  is therefore  $p\mathbf{n}$ , where  $\mathbf{n}$  is the normal vector. The external virtual work component is

$$\delta W_{ext}^p(\phi, \delta\mathbf{v}) = \int_a p\mathbf{n} \cdot \delta\mathbf{v} da. \quad (2.225)$$

In this equation the magnitude of the area element and the orientation of the normal are both displacement-dependent. This proposition means that any change in geometry will result in a change in the equilibrium condition forcing the emergence of a stiffness term.

Although it could be tempting to linearize Equation 2.225 directly, a more direct approach can be done by using a parameterization of the surface as shown in Figure 2.25. Considering the surface area  $a$  to be a single isoparametric element, the normal and area elements are obtained as

$$\mathbf{n} = \frac{\frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta}}{\left| \frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta} \right|}; \quad da = \left| \frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta} \right| d\xi d\eta. \quad (2.226)$$

The normal and area element terms obtained above allow Equation 2.225 to be expressed in the parameter plane as

$$\delta W_{ext}^p(\phi, \delta \mathbf{v}) = \int_{A_\xi} p \delta \mathbf{v} \cdot \left( \frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta} \right) d\xi d\eta. \quad (2.227)$$

Note that the only displacement-dependent terms are the vectors  $\partial \mathbf{x} / \partial \xi$  and  $\partial \mathbf{x} / \partial \eta$ , which linearize into  $\partial \mathbf{u} / \partial \xi$  and  $\partial \mathbf{u} / \partial \eta$  respectively. Hence, the use of the product rule and the cyclic property of the triple product<sup>14</sup> gives

$$D\delta W_{ext}(\phi, \delta \mathbf{v})[\mathbf{u}] = \int_{A_\xi} p \left[ \frac{\partial \mathbf{x}}{\partial \xi} \cdot \left( \frac{\partial \mathbf{u}}{\partial \eta} \times \delta \mathbf{v} \right) - \frac{\partial \mathbf{x}}{\partial \eta} \cdot \left( \frac{\partial \mathbf{u}}{\partial \xi} \times \delta \mathbf{v} \right) \right] d\xi d\eta. \quad (2.228)$$

It is clear that Equation 2.228 is unsymmetric in the sense that the terms  $\mathbf{u}$  and  $\delta \mathbf{v}$  cannot be interchanged without altering the result of the integral. The discretization would yield an unsymmetric tangent matrix component. Assuming the position of points along the boundary  $\partial a$  is fixed or prescribed implies that both the iterative displacement  $\mathbf{u}$  and the virtual velocity  $\delta \mathbf{v}$  are zero a priori along  $\partial A_\xi$  [14]. In addition, if a symmetry plane bisects the region  $a$ , then it is possible to achieve a symmetric expression for  $D\delta W_{ext}^p(\phi, \delta \mathbf{v})[\mathbf{u}]$  as

$$\begin{aligned} D\delta W_{ext}^p(\phi, \delta \mathbf{v})[\mathbf{u}] &= \frac{1}{2} \int_{A_\xi} p \frac{\partial \mathbf{x}}{\partial \xi} \cdot \left[ \left( \frac{\partial \mathbf{u}}{\partial \eta} \times \delta \mathbf{v} \right) + \left( \frac{\partial \delta \mathbf{v}}{\partial \eta} \times \mathbf{u} \right) \right] d\xi d\eta \\ &\quad - \frac{1}{2} \int_{A_\xi} p \frac{\partial \mathbf{x}}{\partial \eta} \cdot \left[ \left( \frac{\partial \mathbf{u}}{\partial \xi} \times \delta \mathbf{v} \right) + \left( \frac{\partial \delta \mathbf{v}}{\partial \xi} \times \mathbf{u} \right) \right] d\xi d\eta. \end{aligned} \quad (2.229)$$

## 2.7. Discretization & Solution

Discretization corresponds to the last step in the process. As it was said in the last section, the spatial linearized equilibrium equations will be used to derive the corresponding discretized equilibrium equations and the tangent matrix. Deriving the terms of the stiffness matrix would require to separate the matrix into three different terms called constitutive, initial stress or geometric, and pressure components.

Once the governing equations are discretized, the Newton-Raphson solution technique will be introduced. Eventually, in order to achieve the convergency in some special cases, the arc length method will be explained.

---

<sup>14</sup> $a \cdot (b \times c) = b \cdot (c \times a) = -c \cdot (b \times a)$ .

### 2.7.1. Discretized Kinematics

The discretization is established in the initial configuration using isoparametric elements, defining the initial position of the element node as

$$\mathbf{X} = \sum_{a=1}^n N_a(\xi_1, \xi_2, \xi_3) \mathbf{x}_a. \quad (2.230)$$

During the motion, nodes and elements are permanently attached to the material particles with which they were initially associated. Consequently, the subsequent motion is fully described in terms of the current position of the nodal particles  $\mathbf{x}_a(t)$  as

$$\mathbf{x} = \sum_{a=1}^n N_a \mathbf{x}_a. \quad (2.231)$$

Differentiating Equation 2.231 with respect to time, the real and virtual velocity are written as

$$\mathbf{v} = \sum_{a=1}^n N_a \mathbf{v}_a; \quad \delta\mathbf{v} = \sum_{a=1}^n N_a \delta\mathbf{v}_a. \quad (2.232)$$

Restricting the motion by an arbitrary increment  $\mathbf{u}$  to be consistent with Equation 2.231 implies that the displacement is interpolated as

$$\mathbf{u} = \sum_{a=1}^n N_a \mathbf{u}_a. \quad (2.233)$$

The fundamental deformation gradient tensor  $\mathbf{F}$  is interpolated over an element by differentiating Equation 2.231 with respect to the initial coordinates, and using the product rule<sup>15</sup> give

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla_0 \mathbf{x} = \nabla(N_a \mathbf{x}_a), \quad (2.234)$$

eventually, noting that the term  $N_a \nabla_0 \mathbf{x}_a(t)$  vanishes, we get

$$\mathbf{F} = \sum_{a=1}^n \mathbf{x}_a \otimes \nabla_0 N_a, \quad (2.235)$$

where  $\nabla_0 N_A = \partial N_a / \partial \mathbf{X}$  can be related to  $\nabla_\xi N_a = \partial N_a / \partial \xi$  by using the chain rule and Equation 2.230, to give

$$\frac{\partial N_a}{\partial \xi} = \left( \frac{\partial \mathbf{X}}{\partial \xi} \right)^T \frac{\partial N_a}{\partial \mathbf{X}}; \quad \frac{\partial N_a}{\partial \mathbf{X}} = \left( \frac{\partial \mathbf{X}}{\partial \xi} \right)^{-T} \frac{\partial N_a}{\partial \xi}. \quad (2.236)$$

which can be expressed as

---

<sup>15</sup> $\nabla(f\mathbf{v}) = f\nabla\mathbf{v} + \mathbf{v} \otimes \nabla f$ .

$$\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \sum_{a=1}^n \mathbf{x}_a \otimes \nabla_{\boldsymbol{\xi}} N_a. \quad (2.237)$$

Equations 2.235 and 2.237 are sufficiently fundamental to justify expansion in detail in order to facilitate their eventual programming. To this effect, these equations are written in an explicit matrix form as

$$\mathbf{F} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix}; \quad F_{iJ} = \sum_{a=1}^n x_{a,i} \frac{\partial N_a}{\partial X_J}; \quad (2.238)$$

$$\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \partial X_1 / \partial \xi_1 & \partial X_1 / \partial \xi_2 & \partial X_1 / \partial \xi_3 \\ \partial X_2 / \partial \xi_1 & \partial X_2 / \partial \xi_2 & \partial X_2 / \partial \xi_3 \\ \partial X_3 / \partial \xi_1 & \partial X_3 / \partial \xi_2 & \partial X_3 / \partial \xi_3 \end{bmatrix}; \quad \frac{\partial X_I}{\partial \xi_\alpha} = \sum_{a=1}^n X_{a,I} \frac{\partial N_a}{\partial \xi_\alpha}. \quad (2.239)$$

From Equation 2.235 further strain magnitudes such as the right and left Cauchy-Green tensors  $\mathbf{C}$  and  $\mathbf{b}$  can be obtained as

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} = \sum_{a,b} (\mathbf{x}_a \cdot \mathbf{x}_b) \nabla_0 N_a \otimes \nabla_0 N_b; \quad C_{IJ} = \sum_{k=1}^3 F_{kI} F_{kJ}; \quad (2.240)$$

$$\mathbf{b} = \mathbf{F} \mathbf{F}^T = \sum_{a,b} (\nabla_0 N_a \cdot \nabla_0 N_b) \mathbf{x}_a \otimes \mathbf{x}_b; \quad b_{ij} = \sum_{K=1}^3 F_{iK} F_{jK}. \quad (2.241)$$

The discretization of the real (or virtual) rate of deformation tensor and the linear strain tensor can be obtained by using Equation 2.232 and into the definition of  $\mathbf{d}$  and Equation 2.233 into Equation 2.212 to give

$$\mathbf{d} = \frac{1}{2} \sum_{a=1}^n (\mathbf{v}_a \otimes \nabla N_a + \nabla N_a \otimes \mathbf{v}_a); \quad (2.242)$$

$$\delta \mathbf{d} = \frac{1}{2} \sum_{a=1}^n (\delta \mathbf{v}_a \otimes \nabla N_a + \nabla N_a \otimes \delta \mathbf{v}_a); \quad (2.243)$$

$$\boldsymbol{\varepsilon} = \frac{1}{2} \sum_{a=1}^n (\mathbf{u}_a \otimes \nabla N_a + \nabla N_a \otimes \mathbf{u}_a). \quad (2.244)$$

where as in Equation 2.237  $\nabla N_a = \partial N_a / \partial \mathbf{x}$  can be obtained from the derivatives of the shape functions with respect to the isoparametric coordinates as

$$\frac{\partial N_a}{\partial \mathbf{x}} = \left( \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)^{-T} \frac{\partial N_a}{\partial \boldsymbol{\xi}}; \quad \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \sum_{a=1}^n \mathbf{x}_a \otimes \nabla_{\boldsymbol{\xi}} N_a, \quad (2.245)$$

and the matrix form definition

$$\frac{\partial x_i}{\partial \xi_\alpha} = \sum_{a=1}^n x_{a,i} \frac{\partial N_a}{\partial \xi_\alpha}. \quad (2.246)$$

Although Equations 2.242 will eventually be expressed in a standard matrix form, if necessary the component tensor products can be expanded in a manner entirely analogous to Equations 2.235 and 2.237.

### 2.7.2. Discretized Equilibrium Equations

In order to obtain the discretized spatial equilibrium equations, recall the spatial virtual work equation given as the total virtual work done by the residual force  $\mathbf{r}$  as

$$\delta W(\phi, \delta \mathbf{v}) = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv - \int_v \mathbf{f} \cdot \delta \mathbf{v} dv - \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} da. \quad (2.247)$$

At this stage it is easier to consider the contribution to  $\delta W(\phi, \delta \mathbf{v})$  caused by a single virtual nodal velocity  $\delta \mathbf{v}_a$  occurring at a typical node  $a$  of element  $(e)$ . Introducing the interpolation for  $\delta \mathbf{v}$  (2.232) and  $\delta \mathbf{d}$  (2.243) gives

$$\delta W^{(e)}(\phi, N_a \delta \mathbf{v}_a) = \int_{v^{(e)}} \boldsymbol{\sigma} : (\delta \mathbf{v}_a \otimes \nabla N_a) dv - \int_{v^{(e)}} \mathbf{f} \cdot (N_a \delta \mathbf{v}_a) dv - \int_{\partial v^{(e)}} \mathbf{t} \cdot (N_a \delta \mathbf{v}_a) da. \quad (2.248)$$

Observing that the virtual nodal velocities are independent of the integration, remembering that  $\delta \mathbf{v}$  is assumed constant during an incremental  $\mathbf{u}(x)$ , and a product property of the double contraction<sup>16</sup>, enables the summation to be rearranged to give

$$\delta W^{(e)}(\phi, N_a \delta \mathbf{v}_a) = \delta \mathbf{v}_a \cdot \left( \int_{v^{(e)}} \boldsymbol{\sigma} \nabla N_a dv - \int_{v^{(e)}} N_a \mathbf{f} dv - \int_{\partial v^{(e)}} N_a \mathbf{t} da \right). \quad (2.249)$$

The virtual work per element  $(e)$  per node  $a$  can, alternatively, be expressed in terms of the internal and external equivalent nodal forces  $\mathbf{T}_a^{(e)}$  and  $\mathbf{F}_a^{(e)}$  as

$$\delta W(\phi, N_a \delta \mathbf{v}_a) = \delta \mathbf{v}_a \cdot (\mathbf{T}_a^{(e)} - \mathbf{F}_a^{(e)}), \quad (2.250)$$

where the latter expression is simply velocity times force, and the internal and external equivalent nodal forces are defined as

$$\mathbf{T}_a^{(e)} = \int_{v^{(e)}} \boldsymbol{\sigma} \nabla N_a dv; \quad T_{a,i}^{(e)} \sum_{j=1}^3 \int_{v^{(e)}} \sigma_{ij} \frac{\partial N_a}{\partial x_j} dv; \quad (2.251)$$

$$\mathbf{F}_a^{(e)} = \int_{v^{(e)}} N_a \mathbf{f} dv + \int_{\partial v^{(e)}} N_a \mathbf{t} da. \quad (2.252)$$

---

<sup>16</sup> $\boldsymbol{\sigma} : (\mathbf{u} \otimes \mathbf{v}) = \mathbf{u} \cdot \boldsymbol{\sigma} \mathbf{v}$  for any vectors  $\mathbf{u}, \mathbf{v}$ .

In this equation the Cauchy stress  $\sigma$  is found from the appropriate constitutive equation given in Section 2.5, which will involve the calculation of the left Cauchy-Green tensor  $b = \mathbf{F}\mathbf{F}^T$ .

The contribution to  $\delta W(\phi, N_a \delta \mathbf{v}_a)$  from all elements  $e$  (1 to  $m$ ) containing node  $a$  ( $e \ni a$ ) is

$$\delta W(\phi, N_a \delta \mathbf{v}_a) = \sum_{\substack{e=1 \\ e \ni a}}^m \delta W^{(e)}(\phi, N_a \delta \mathbf{v}_a) = \delta \mathbf{v}_a \cdot (\mathbf{T}_a - \mathbf{F}_a), \quad (2.253)$$

where the assembled equivalent nodal forces are

$$\mathbf{T}_a = \sum_{\substack{e=1 \\ e \ni a}}^m \mathbf{T}_a^{(e)}; \quad \mathbf{F}_a = \sum_{\substack{e=1 \\ e \ni a}}^m \mathbf{F}_a^{(e)}. \quad (2.254)$$

Finally, the contribution to  $\delta W(\phi, \delta \mathbf{v})$  from all nodes  $n$  in the finite element mesh is

$$\delta W(\phi, \delta \mathbf{v}) = \sum_{a=1}^N \delta W(\phi, N_a \delta \mathbf{v}_a) = \sum_{a=1}^N \delta \mathbf{v}_a \cdot (\mathbf{T}_a - \mathbf{F}_a) = 0. \quad (2.255)$$

Because the virtual work equation must be satisfied for any arbitrary virtual nodal velocities, the discretized equilibrium equations, in terms of the nodal residual force  $\mathbf{R}_a$ , emerge as,

$$\mathbf{R}_a = \mathbf{T}_a - \mathbf{F}_a = 0. \quad (2.256)$$

Consequently, the equivalent internal nodal forces are in equilibrium with the equivalent external forces at each node.

For convenience, these equivalent forces are assembled into single arrays to define the complete internal and external forces  $\mathbf{T}$  and  $\mathbf{F}$  respectively, as well as the complete residual force  $\mathbf{R}$  as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_n \end{bmatrix}; \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_n \end{bmatrix}; \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_n \end{bmatrix}. \quad (2.257)$$

These definitions enable the discretized virtual work Equation 2.255 to be rewritten as

$$\delta W(\phi, \delta \mathbf{v}) = \delta \mathbf{v}^T \mathbf{R} = \delta \mathbf{v}^T (\mathbf{T} - \mathbf{F}) = 0, \quad (2.258)$$

where  $\delta \mathbf{v}$  is the complete virtual velocity vector. Finally, recalling that the internal equivalent forces are nonlinear functions of the current nodal positions  $\mathbf{x}_a$  and defining a complete vector of unknowns  $\mathbf{x}$  as the array containing all nodal positions as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \quad (2.259)$$

enables the complete nonlinear equilibrium equations to be symbolically assembled as

$$\mathbf{R}(\mathbf{x}) = \mathbf{T}(\mathbf{x}) - \mathbf{F}(\mathbf{x}) = \mathbf{0}. \quad (2.260)$$

This discretized expression corresponds to the weak form of the differential equilibrium equation  $\mathbf{r} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}$ , seen in previous sections.

### 2.7.3. Discretization of the Linearized Equilibrium Equations

Equation 2.260 represents a set of nonlinear equilibrium equations with the current nodal positions as unknowns. The solution of these equations is achieved using a Newton-Raphson iterative procedure that involves the discretization of the linearized equilibrium equations. For notational convenience the virtual work equation is split into internal and external work components as

$$\delta W(\phi, \delta \mathbf{v}) = \delta W_{int}(\phi, \delta \mathbf{v}) - \delta W_{ext}(\phi, \delta \mathbf{v}). \quad (2.261)$$

It is worth reiterating the general discussion of Section 2.6.1 to inquire in more detail why the discretization of the linearized equilibrium Equation 2.202 basically yield a tangent stiffness matrix. Recalling equation  $\delta W(\phi, N_a \delta \mathbf{v}_a) = \delta \mathbf{v}_a \cdot (\mathbf{T}_a^{(e)} - \mathbf{F}_a^{(e)})$ , essentially expresses the contribution of the nodal equivalent forces  $\mathbf{T}_a^{(e)}$  and  $\mathbf{F}_a^{(e)}$  to the overall equilibrium of node  $a$ . Observing that  $\mathbf{F}_a^{(e)}$  may be position-dependent, linearization of the following equation

$$\mathbf{F}_a^{(e)} = \int_{v^{(e)}} N_a \mathbf{f} dv + \int_{\partial v^{(e)}} N_a \mathbf{t} da, \quad (2.262)$$

in the direction  $N_b \mathbf{u}_b$ , with  $N_a \delta \mathbf{v}_a$  remaining constant, expresses the change in the nodal equivalent forces  $\mathbf{T}_a^{(e)}$  and  $\mathbf{F}_a^{(e)}$ , at node  $a$ , due to a change  $\mathbf{u}_b$  in the current position of node  $b$  as

$$\begin{aligned} D\delta W^{(e)}(\phi, N_a \delta \mathbf{v}_a)[N_b \mathbf{u}_b] &= D(\delta \mathbf{v}_a \cdot (\mathbf{T}_a^{(e)} - \mathbf{F}_a^{(e)}))[N_b \mathbf{u}_b] \\ &= \delta \mathbf{v}_a \cdot D(\mathbf{T}_a^{(e)} - \mathbf{F}_a^{(e)})[N_b \mathbf{u}_b] \\ &= \delta \mathbf{v}_a \cdot \mathbf{K}_{ab}^{(e)} \mathbf{u}_b. \end{aligned} \quad (2.263)$$

The relationship between changes in forces at node  $a$  due to changes in the current position of node  $b$  is furnished by the tangent stiffness matrix  $\mathbf{K}_{ab}^{(e)}$ . In physical terms the tangent stiffness provides the Newton-Raphson procedure with the operator that adjust current nodal positions so that the deformation-dependent equivalent nodal forces tend toward being in equilibrium with the external equivalent nodal forces.

### 2.7.3.1. Constitutive Component

Although a matrix form of every component of the stiffness matrix can be derived, here the indicial form is preferred since it is easier to implement later in the computer. The constitutive contribution to the linearized virtual work Equation 2.222 for element ( $e$ ) linking nodes  $a$  and  $b$  is

$$\begin{aligned} & D\delta W_c^{(e)}(\phi, N_a \delta \mathbf{v}_a) [N_b \mathbf{u}_b] \\ &= \int_{v^{(e)}} \frac{1}{2} (\delta \mathbf{v}_a \otimes \nabla N_a + \nabla N_a \otimes \delta \mathbf{v}_a) : \mathbf{c} : \frac{1}{2} (\mathbf{u}_b \otimes \nabla N_b + \nabla N_b \otimes \mathbf{u}_b) dv. \end{aligned} \quad (2.264)$$

This equation can be rewritten in indicial notation as

$$\begin{aligned} & D\delta W_c^{(e)}(\phi, N_a \delta \mathbf{v}_a) [N_b \mathbf{u}_b] \\ &= \sum_{i,j,k,l=1}^3 \int_{v^{(e)}} \frac{1}{2} \left( \delta v_{a,i} \frac{\partial N_a}{\partial x_j} + \delta v_{a,j} \frac{\partial N_a}{\partial x_i} \right) c_{ijkl} \frac{1}{2} \left( u_{b,k} \frac{\partial N_b}{\partial x_l} + u_{b,l} \frac{\partial N_b}{\partial x_k} \right) dv \\ &= \sum_{i,j,k,l=1}^3 \delta v_{a,i} \left( \int_{v^{(e)}} \frac{\partial N_a}{\partial x_j} c_{ijkl} \frac{\partial N_b}{\partial x_l} dv \right) u_{b,k} \\ &= \delta \mathbf{v}_a \cdot \mathbf{K}_{c,ab}^{(e)} \mathbf{u}_b. \end{aligned} \quad (2.265)$$

where the constitutive component of the tangent matrix relating node  $a$  to node  $b$  in element ( $e$ ) is

$$[\mathbf{K}_{c,ab}]_{ij} = \int_{v^{(e)}} \sum_{k,l=1}^3 \frac{\partial N_a}{\partial x_k} c_{ikjl} \frac{\partial N_b}{\partial x_l} dv; \quad i, j = 1, 2, 3 \quad (2.266)$$

### 2.7.3.2. Initial Stress (Geometric) Component

Before discretizing the second term in the linearized equilibrium equation 2.222, it is necessary to interpolate the gradients of  $\delta \mathbf{v}$  and  $\mathbf{u}$  as

$$\nabla \delta \mathbf{v} = \sum_{a=1}^n \delta \mathbf{v}_a \otimes \nabla N_a; \quad (2.267)$$

$$\nabla \mathbf{u} = \sum_{b=1}^n \mathbf{u}_b \otimes \nabla N_b. \quad (2.268)$$

Introducing these expressions in the second term of Equation 2.222 and recalling 16 for  $\sigma$  enables the initial stress contribution to the linearized virtual work equation for element ( $e$ ) linking nodes  $a$  and  $b$  to be found as

$$\begin{aligned}
D\delta W_\sigma(\phi, N_a \delta \mathbf{v}_a) [N_b \mathbf{u}_b] &= \int_v \boldsymbol{\sigma} : [(\nabla \mathbf{u}_b)^T \nabla \delta \mathbf{v}_a] dv \\
&= \int_{v^{(e)}} \boldsymbol{\sigma} : [(\delta \mathbf{v}_a \cdot \mathbf{u}_b) \nabla N_b \otimes \nabla N_a] dv \\
&= (\delta \mathbf{v}_a \cdot \mathbf{u}_b) \int_{v^{(e)}} \nabla N_a \cdot \boldsymbol{\sigma} \nabla N_b dv. \tag{2.269}
\end{aligned}$$

Observing that the integral in Equation 2.269 is a scalar, and noting that  $\delta \mathbf{v}_a \cdot \mathbf{u}_b = \delta \mathbf{v}_a \cdot \mathbf{I} \mathbf{u}_b$ , the expression can be rewritten in matrix form as

$$D\delta W_\sigma(\phi, N_a \delta \mathbf{v}_a) [N_b \mathbf{u}_b] = \delta \mathbf{v}_a \cdot \mathbf{K}_{\sigma,ab}^{(e)} \mathbf{u}_b, \tag{2.270}$$

where the component of the so-called initial stress matrix is

$$[\mathbf{K}_{\sigma,ab}]_{ij} = \int_{v^{(e)}} \sum_{k,l=1}^3 \frac{\partial N_a}{\partial x_k} \sigma_{kl} \frac{\partial N_b}{\partial x_l} \delta_{ij} dv; \quad i, j = 1, 2, 3 \tag{2.271}$$

#### 2.7.3.3. External Force Component

The body forces are invariably independent of the motion and consequently do not contribute to the linearized work. However, for the particular case of enclosed normal pressure, the linearization of the virtual work is given by Equation 2.229 as<sup>17</sup>

$$\begin{aligned}
D\delta W_{ext}^p(\phi, \delta \mathbf{v}) [\mathbf{u}] &= \frac{1}{2} \int_{A_\xi} p \frac{\partial \mathbf{x}}{\partial \xi} \cdot \left[ \left( \frac{\partial \delta \mathbf{v}}{\partial \eta} \times \mathbf{u} \right) - \left( \delta \mathbf{v} \times \frac{\partial \mathbf{u}}{\partial \eta} \right) \right] d\xi d\eta \\
&\quad - \frac{1}{2} \int_{A_x i} p \frac{\partial \mathbf{x}}{\partial \eta} \cdot \left[ \left( \frac{\partial \delta \mathbf{v}}{\partial \xi} \times \mathbf{u} \right) - \left( \delta \mathbf{v} \times \frac{\partial \mathbf{u}}{\partial \xi} \right) \right] d\xi d\eta. \tag{2.272}
\end{aligned}$$

Similar surface interpolations, such as the implicit done in  $\mathbf{x}(\xi, \eta) = \sum_a N_a \mathbf{x}_a$ , are performed to both  $\delta \mathbf{v}$  and  $\mathbf{u}$  in Equation 2.272 to give

$$\begin{aligned}
D\delta W_{ext}^{p(e)}(\phi, N_a \delta \mathbf{v}_a) [N_b \mathbf{u}_b] &= (\delta \mathbf{v}_a \times \mathbf{u}_b) \cdot \frac{1}{2} \int_{A_\xi} p \frac{\partial \mathbf{x}}{\partial \xi} \left( \frac{\partial N_a}{\partial \eta} N_b - \frac{\partial N_b}{\partial \eta} N_a \right) d\xi d\eta \\
&\quad - (\delta \mathbf{v}_a \times \mathbf{u}_b) \cdot \frac{1}{2} \int_{A_\xi} p \frac{\partial \mathbf{x}}{\partial \eta} \left( \frac{\partial N_a}{\partial \xi} N_b - \frac{\partial N_b}{\partial \xi} N_a \right) d\xi d\eta \\
&= (\delta \mathbf{v}_a \times \mathbf{u}_b) \cdot \mathbf{k}_{p,ab}, \tag{2.273}
\end{aligned}$$

---

<sup>17</sup>The minus sign inside the brackets in Equation 2.229 appears because of the change in the order of the cross product.

where  $\mathbf{k}_{p,ab}$  is the so-called vector of stiffness coefficients. Equation 2.273 can be expressed as a term of the tangent matrix as

$$D\delta W_{ext}^p(\phi, \delta \mathbf{v})[\mathbf{u}] = \delta \mathbf{v}_a \cdot \mathbf{K}_{p,ab}^{(e)} \mathbf{u}_b, \quad (2.274)$$

and where the external pressure component of the tangent matrix is

$$\mathbf{K}_{p,ab}^{(e)} = \mathcal{E} \mathbf{k}_{p,ab}^{(e)}; \quad [\mathbf{K}_{p,ab}^{(e)}]_{ij} = \sum_{k=1}^3 \mathcal{E}_{ijk} [\mathbf{k}_{p,ab}^{(e)}]_k; \quad i, j = 1, 2, 3 \quad (2.275)$$

the  $\mathcal{E}$  tensor is the so-called *Levi-Civita* tensor.

#### 2.7.3.4. Tangent Matrix

The discretized virtual work equations of  $\mathbf{K}_c$  (2.265),  $\mathbf{K}_\sigma$  (2.270) and  $\mathbf{K}_p$  (2.274), for an element ( $e$ ) linking nodes  $a$  and  $b$ , can be expressed jointly in terms of the total stiffness matrix  $\mathbf{K}_{ab}$  as

$$D\delta W^{(e)}(\phi, N_a \delta \mathbf{v}_a)[N_b \mathbf{u}_b] = \delta \mathbf{v}_a \cdot \mathbf{K}_{ab}^{(e)} \mathbf{u}_b; \quad (2.276)$$

$$\mathbf{K}_{ab}^{(e)} = \mathbf{K}_{c,ab}^{(e)} + \mathbf{K}_{\sigma,ab}^{(e)} - \mathbf{K}_{p,ab}^{(e)}. \quad (2.277)$$

Equation 2.258 allows to express the element assembly process by using the complete virtual velocity vector together with the corresponding nodal displacements and the assembled tangent stiffness matrix  $\mathbf{K}$  to yield

$$D\delta W(\phi, \delta \mathbf{v})[\mathbf{u}] = \delta \mathbf{v}^T \mathbf{K} \mathbf{u}, \quad (2.278)$$

where the tangent stiffness matrix is defined as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \dots & \mathbf{K}_{1n} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \dots & \mathbf{K}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \mathbf{K}_{n2} & \dots & \mathbf{K}_{nn} \end{bmatrix}. \quad (2.279)$$

#### 2.7.4. Numerical Methods

Nonlinear structural problems are generally solved using iterative methods, such as Newton-Raphson, line search method, arc length method, etc. Such algorithms are successful for solving the nonlinear equilibrium equations of the model. However, the computing time is usually large as compared to a linear solution, due to both the number of iterations needed and the computation of tangent stiffness matrices.

The two methods addressed here are Newton-Raphson and arc-length method. Both methods are incremental-iterative schemes, having two well-differentiated parts: the *increment*

part and the iteration *part*. The former controls the step size of a variable (force or displacement) in order to find the solution, whereas the latter, which is found inside the *while loop* (See Algorithm 1, line 7), seeks to decrease the difference between the unbalanced internal  $\mathbf{T}$  and external forces  $\mathbf{F}$ . It takes several iterations to determine an acceptable solution to a given increment.

#### 2.7.4.1. Newton-Raphson

The nonlinear equilibrium set of equations to solve is

$$\mathbf{R}(\mathbf{x}) = \mathbf{T}(\mathbf{x}) - \mathbf{F} = 0. \quad (2.280)$$

It has quadratic convergence and it is the easiest method to implement. The solution is found by applying specified loads gradually and incrementally working towards the final solution. Therefore, the external force is “broken” into a number of load increments to find the approximate equilibrium solution at the end of each load increment.

The Newton-Raphson method comes natural to solve the linearized virtual work equation. Equation 2.201 is described in Section 2.6 as

$$\delta W(\phi_k, \delta \mathbf{v}) + D\delta W(\phi_k, \delta \mathbf{v})[\mathbf{u}] = 0. \quad (2.281)$$

This latter equation is linearized recalling that

$$\delta W(\phi, \delta \mathbf{v}) = \delta \mathbf{v}^T \mathbf{R}; \quad (2.282)$$

$$D\delta W(\phi, \delta \mathbf{v})[\mathbf{u}] = \delta \mathbf{v}^T \mathbf{Ku}. \quad (2.283)$$

Because the nodal virtual velocities are arbitrary, the discretized Newton-Raphson Equation 2.281 leads to

$$\begin{aligned} \mathbf{R}(\mathbf{x}_{k+1}) &\approx \mathbf{R}(\mathbf{x}_k) + D\mathbf{R}(\mathbf{x}_k)[\mathbf{u}] = 0; \\ \mathbf{R}(\mathbf{x}_{k+1}) &\approx \mathbf{R}(\mathbf{x}_k) + \mathbf{K}(\mathbf{x}_k) \cdot \mathbf{u} = 0. \end{aligned} \quad (2.284)$$

The equation above explains the etymology of *tangent* stiffness matrix, which is due to its definition as the directional derivative of the residual function, and thus tangent to  $\mathbf{R}(\mathbf{x}_k)$  (see Figure 2.26). The tangent stiffness matrix is expressed as

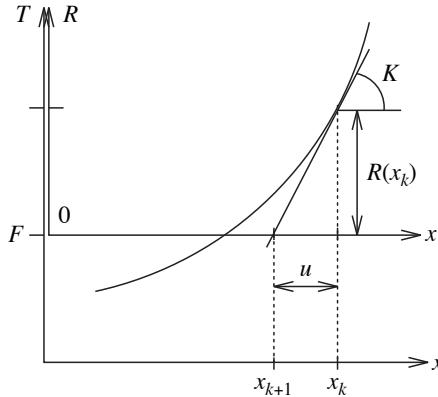
$$\mathbf{K} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \quad (2.285)$$

From Equation 2.284, the displacement  $\mathbf{u}$  is isolated as

$$\mathbf{u} = -\mathbf{K}(\mathbf{x}_k)^{-1} \mathbf{R}(\mathbf{x}_k). \quad (2.286)$$

Given a solution estimate  $\mathbf{x}_k$  at iteration  $k$ , the Newton-Raphson algorithm computes a new value  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}$  by establishing the linear approximation as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}. \quad (2.287)$$



**Figure 2.26:** Newton-Raphson iteration for a one-degree-of-freedom case [6]. The tangent line to the function  $\mathbf{R}(\mathbf{x})$ , when  $\mathbf{x} = \mathbf{x}_k$ , intersects  $\mathbf{R} = 0$ , finds the new solution estimate  $\mathbf{x}_{k+1}$ , being  $\mathbf{K}$  the slope of the tangent line.

Once the solution of iteration  $k$  has converged a new load increment is set. The external load  $\mathbf{F}$  is defined as

$$\mathbf{F} = \sum_{i=1}^l \Delta \mathbf{F}_i, \quad (2.288)$$

where  $l$  is the total number of load increments. The more increments taken, the easier it becomes to find a converged solution for each individual load step [15]. An outline of the complete solution algorithm is shown in 1.

---

**Algorithm 1:** Newton-Raphson

---

**Input:** geometry, material properties, and solution parameters

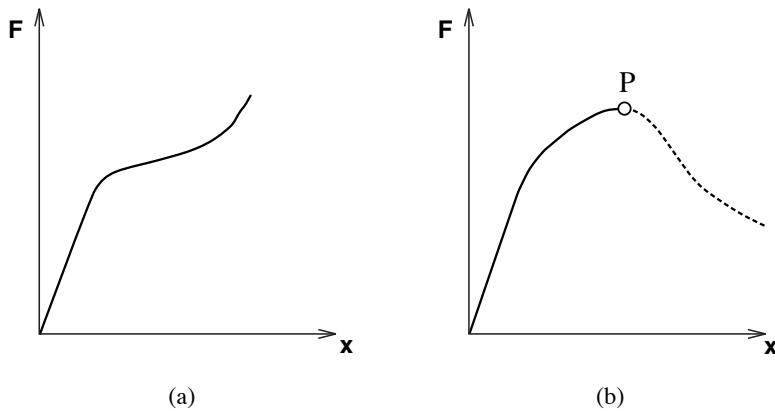
```

1 Initialize  $\mathbf{F} = \mathbf{0}$ ,  $\mathbf{x} = \mathbf{X}$ ,  $\mathbf{R} = \mathbf{0}$ 
2 for  $incrm = 1 : nincr$  do
3   Find  $\Delta \mathbf{F}$ 
4   Set  $\mathbf{F} = \mathbf{F} + \Delta \mathbf{F}$ 
5   Set  $\mathbf{R} = \mathbf{R} - \Delta \mathbf{F}$ 
7   while  $\|\mathbf{R}\| / \|\mathbf{F}\| >$  do
8     Find  $\mathbf{K}$ 
9     Solve  $\mathbf{Ku} = -\mathbf{R}$ 
10    Update  $\mathbf{x} = \mathbf{x} + \mathbf{u}$ 
11    Find  $\mathbf{F}, \mathbf{b}$  and  $\sigma$ 
12    Find  $\mathbf{T}$ 
13    Find  $\mathbf{R} = \mathbf{T} - \mathbf{F}$ 
14  end
15 end
```

---

Nevertheless it will be seen in Chapter 4 that in some cases do not succeed. The load-displacement response diagram in Figure 2.27 (a) shows the typical response or *equilibrium*

*path* of a structure. In this case, the Newton-Raphson method does not have any inconvenient to find a solution since it is always able to find a tangent line to the equilibrium path. However, in Figure 2.27 (b) appears a *limit point P*. This limit point is found when the tangent to the equilibrium path is horizontal. The equilibrium path before **P** is called primary, while the path behind **P** known as secondary. It is beyond **P** where Newton-Raphson method is not capable to converge. A modification<sup>18</sup> in the Newton-Raphson method would lead to solve the problem of limit points, but regrettably it is not able to find a solution in a *snap-back* behaviour (Figure 2.28 (a)).



**Figure 2.27:** Response diagrams: (a) typical load-deflection diagram showing equilibrium path; (b) diagram distinguishing primary from secondary equilibrium path.

#### 2.7.4.2. Arc Length Method

The arc-length method solves the problem caused by the snap-back behaviour. When the Newton-Raphson scheme finds a limit point, by increasing the load, the solution experience convergency problems. For instance, by looking into Figure 2.28 from point **A** it may jump to the alternate equilibrium position **A'**. Sometimes instead of controlling the load parameter, it is the displacement the variable which is prescribed. However, this technique makes the solution to progress erroneously from **B** to **B'**. Therefore, the arc-length method allows to find equilibrium path by controlling both the load and displacement depending on the results in the current iteration.

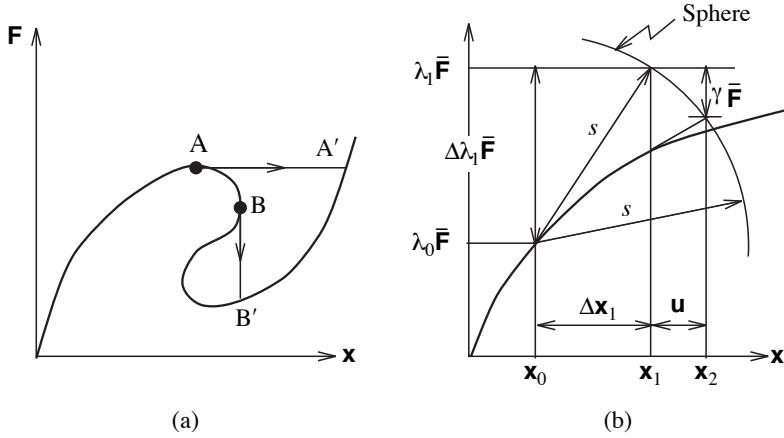
Since the method treats the load factor as a variable, it becomes an additional unknown in equilibrium equations. Simplification of the constraint equation leads to a quadratic equation, whose roots are used for determining the load factor. The arc-length method controls the magnitude of the external loads by the load-level parameter  $\lambda$  so that the equilibrium equation becomes

$$\mathbf{R}(\mathbf{x}, \lambda) = \mathbf{T}(\mathbf{x}) - \lambda \mathbf{F} = 0. \quad (2.289)$$

The arc-length method is aimed to find the intersection of Equation 2.289 with a constant  $s$  termed as the arc-length (see Figure 2.28 (b)), and can be written in differential form as

---

<sup>18</sup>This modification consist of finding the smallest eigenvalue of  $\mathbf{K}$ . This value plus  $\mathbf{I}$  is added to  $\mathbf{K}$ . This allows the tangent line to have a negative slope.



**Figure 2.28:** (a) Snap-back; (b) Spherical arc-length method [15].

$$s = \int \sqrt{d\mathbf{x}^T d\mathbf{x} + d\lambda^2 \Psi^2 \mathbf{F}^T \mathbf{F}}, \quad (2.290)$$

or in increment form as

$$\mathbf{a} = \Delta \mathbf{x}^T \Delta \mathbf{x} + \Delta \lambda^2 \Psi^2 \mathbf{F}^T \mathbf{F} = s^2, \quad (2.291)$$

where  $\Delta \mathbf{x}$  is the vector of incremental displacements,  $\Delta \lambda$  is the incremental load factor,  $s$  is the fixed radius of the desired intersection, and  $\Psi$  is a scaling parameter for loading terms.

After a linearization and simplification process [16], Equations 2.290 and 2.291 give the following system

$$\begin{bmatrix} \delta \mathbf{x} \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{K} & -\mathbf{F} \\ 2\Delta \mathbf{x}^T & 2\Delta \lambda \Psi^2 \mathbf{F} \mathbf{F}^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_{old} \\ \mathbf{a}_{old} \end{bmatrix}, \quad (2.292)$$

where  $\delta \mathbf{x}$  is the iterative change in displacement vector, which is  $\mathbf{u}$ ; and  $\delta \lambda$  is the iterative change in load-factor. In order not to confuse this last parameter with the incremental load factor it will be termed as  $\gamma$ .  $\mathbf{R}_{old}$  and  $\mathbf{a}_{old}$  are the previous values of residual force and arc-length. After the iterative change in  $\mathbf{u}$  and  $\gamma$  is computed, the displacement vector and load factors are updated.

However, instead of solving Equation 2.292 a constraint equation can be introduced to by following the technique of Baltoz and Dhatt [16] for displacement control at a single point. For the new unknown load level

$$\Delta \lambda_{k+1} = \Delta \lambda_k + \gamma; \quad \lambda_{k+1} = \lambda_k + \gamma, \quad (2.293)$$

according to the technique, the iterative change of displacement is written as

$$\mathbf{u} = \mathbf{u}_R + \gamma \mathbf{u}_F, \quad (2.294)$$

where the subindexes  $R$  and  $F$  stand for residual and force, respectively, and these displacements are computed as

$$\mathbf{u}_R = \mathbf{K}(\mathbf{x}_k)^{-1} \mathbf{R}(\mathbf{x}_k, \lambda_k); \quad \mathbf{u}_F = -\mathbf{K}(\mathbf{x}_k)^{-1} \mathbf{F}. \quad (2.295)$$

The incremental displacement for the next increment is expressed as

$$\Delta \mathbf{x}_{k+1} = \Delta \mathbf{x}_k + \mathbf{u}; \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}. \quad (2.296)$$

Substituting Equations 2.294 and 2.296 into the constraint Equation 2.291 yields to

$$a_1\gamma^2 + a_2\gamma + a_3 = 0, \quad (2.297)$$

where

$$a_1 = \mathbf{u}_F^T \mathbf{u}_F + \Psi^2 \mathbf{F} \mathbf{F}^T \quad (2.298)$$

$$a_2 = 2\mathbf{u}_F^T (\Delta \mathbf{x}_k + \mathbf{u}_R) + 2\Delta \lambda_k \Psi^2 \mathbf{F}^T \mathbf{F} \quad (2.299)$$

$$a_3 = \mathbf{u}_R^T (2\Delta \mathbf{x}_k + \mathbf{u}_R) + \Delta \mathbf{x}_k^T \Delta \mathbf{x}_k - s^2 \quad (2.300)$$

There are two solution  $\gamma^{(1)}$  and  $\gamma^{(2)}$  to Equation 2.297, which when substituted into Equations 2.294, 2.295, 2.293 and 2.296 give to generalized vectors  $\mathbf{s}_{k+1}^{(1)}$  and  $\mathbf{s}_{k+1}^{(2)}$ . The correct parameter  $\gamma^{(1)}$  or  $\gamma^{(2)}$ , is that which gives the minimum angle  $\theta$  between  $\mathbf{s}_k$  and  $\mathbf{s}_k^j$  where  $\theta$  is obtained from

$$\cos \theta^{(j)} = \frac{\mathbf{s}_k^T \mathbf{s}_{k+1}^{(j)}}{s^2}; \quad \mathbf{s}_k = \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \lambda_k \Psi \mathbf{F} \end{bmatrix}; \quad \mathbf{s}_{k+1}^{(j)} = \begin{bmatrix} \Delta \mathbf{x}_k + \mathbf{u}^{(j)} \\ (\Delta \lambda_k + \gamma^{(j)} \Psi \mathbf{F}) \end{bmatrix}. \quad (2.301)$$

# CHAPTER 3. COMPUTER IMPLEMENTATION

A FEM software may result in large implementations. Therefore, the use of the Object Oriented Programming (OOP) paradigm allow us to structure the code in a well-defined classes forming a hierarchy. Furthermore, the use of OOP allows to develop a multipurpose software in a natural way; the code expansion is more simple as new implementations have little impact. Matlab is the *scripting*<sup>1</sup> programming language used.

In this chapter the main benefits of developing *FEM-MAT-OO* based on OOP are explained. The overall structure, as well as some important *FEM-MAT-OO* class examples are addressed. At the end of the chapter both test-driven development directive, the guideline towards new implementations and updates; and the control version tool, the responsible of managing the changes in the code, are explained. These topics are recurrent in software development owing to the fact that they constitute the pillar in collaborative codes.

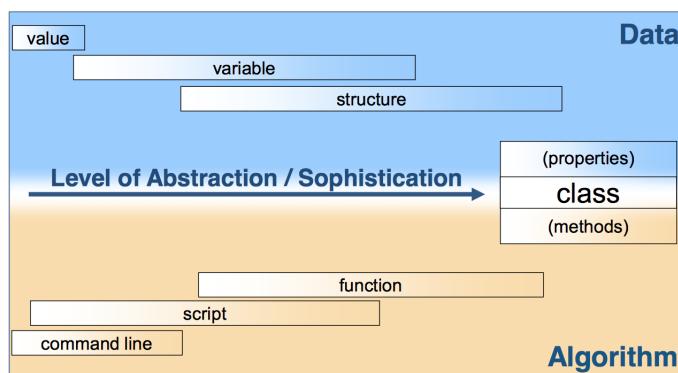
## 3.1. Object Oriented Programming

The finite element method is one of the most widely used numerical tools in engineering, particularly in structural analysis. The reason for the widespread use of the finite element method is that it is both powerful and lends itself easily to modular programming [17].

In order to explain the main elements and principles related to object-oriented programming, *FEM-MAT-OO*-related cases are used.

### 3.1.1. Object-Oriented Elements

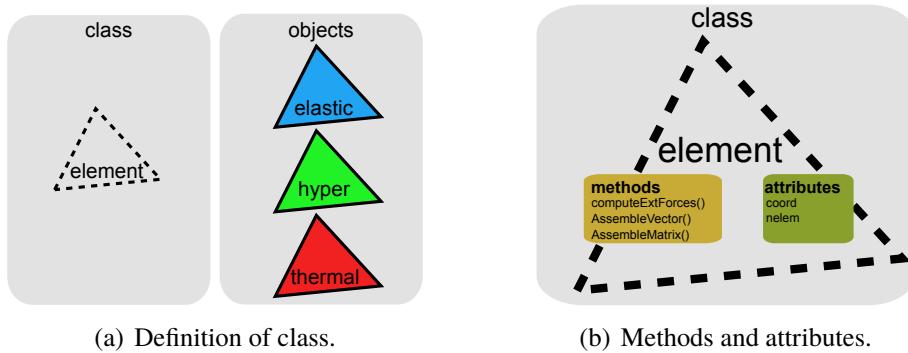
The main elements or *concepts* when explaining OOP are objects, classes, attributes and methods. Figure 3.1 separates code units into two parts: data and algorithm. This concepts, which are listed depending on its level of abstraction, merge encountering the main concept in object-oriented programming: the *class*.



**Figure 3.1:** Definition of a class as the highest level of sophistication [18].

Before defining the class concept, it is convenient to present the *object*. An object is an abstraction of something in a problem domain, reflecting the capabilities of a system to keep information about it, interact with it, or both. A synonym for object is an instance.

<sup>1</sup>A programming language which does not need to require and explicit compilation step are called scripting languages.



**Figure 3.2:** *Element* class as a template to make (instantiate) objects.

The *class* term appears when one or more objects with a uniform set of attributes and services are able to be defined under the same description. Sometimes classes are mislead with *structs*. Although classes and structs have named fields (called *attributes* in classes), classes differ from structs in that classes also encapsulate operations related to them in which are called *methods*. In addition, each class includes a constructor, a special method used to create the object.

*Attributes* (or properties in Matlab) describe the values (state) of the object, to be exclusively manipulated by the services of that object. The attributes are generally the variables of a object, and each attribute has a type. The services of the object are called *methods*. These methods are implemented using functions which are attached to an object, allowing the object to perform actions.

### 3.1.2. Object-Oriented Principles

The key principles of object-oriented programming are abstraction, encapsulation, inheritance and polymorphism. The *abstraction* of a data type is realized by using a class. It is the process of moving from a specific idea to a general one. Sometimes abstraction is confused with *encapsulation*. The difference is subtle due to abstraction is achieved by implementing encapsulation. Encapsulation is basically the presentation of a simple concept (object) to the user by hiding irrelevant details. Encapsulation is achieved by restricting the access only to permitted classes. Abstraction could be seen in Listing 3.1 in which a *triangle\_linear* object is created. Any object of class *Physical\_Problem* could perform *preProcess*, *computeVariables* and *print* methods. Although a user do not know how the code computes the output variables of an element, this user knows that any kind of object is subject to these methods.

```
%% Main.m
triangle_linear = Physical_Problem('TOY_EXAMPLE');
triangle_linear.preProcess;
triangle_linear.computeVariables;
triangle_linear.print;
```

**Listing 3.1:** Main script.

On the contrary, encapsulation is seen in Listing 3.2. Inside *compute\_Hyperelasticity* many different tensors are computed. Each tensor is bundled in a capsule (function). For

instance, for computing the Cauchy stress tensor it is not necessary to see the way the other tensors are computed, and thus is hidden. Moreover, encapsulation is achieved by restricting the access to the objects by using the following property attributes

- **Public.** Allows unrestricted access to the class.
- **Protected.** Access is restricted to the class and subclasses.
- **Private.** Access is only possible from class members (not subclasses).

```

function [ctens,sigma] = compute_Hyperelasticity(obj,coord)
    % Deformation Gradient tensor
    [F,Fjacb] = obj.compute_Deformation_Gradient(coord,obj.cartd0);

    % Right Cauchy Deformation tensor
    Crcg = obj.compute_Right_Cauchy_Deformation(F);

    % Lagrangian Elasticity tensor
    Ctens = obj.compute_Lagrangian_Elasticity(Crcg,Fjacb);

    % Second Piola-Kirchhoff Stress tensor
    S = obj.compute_Second_PK_Stress(Crcg,Fjacb);

    % Eulerian Elasticity & Cauchy Stress tensors
    [ctens,sigma] = obj.pushForward(Ctens,S,F,Fjacb);
end

```

**Listing 3.2:** Element\_Hyperelastic class.

The *inheritance* concept is used to define object hierarchies. An object can have many children which inherit its data and member functions. A subclass is usually of a special type and has additional methods and attributes. Figure 3.3 shows the UML (Unified Modelling Language) diagram related to element classes. As it can be seen the superclass (or parent class) is **Element**. This **Element** class has 3 children which in turn can also have more child classes. The child class **Element\_Hyperelastic** inherits the behaviour of the parent class, that is, the child class is able to use parent's methods and attributes.

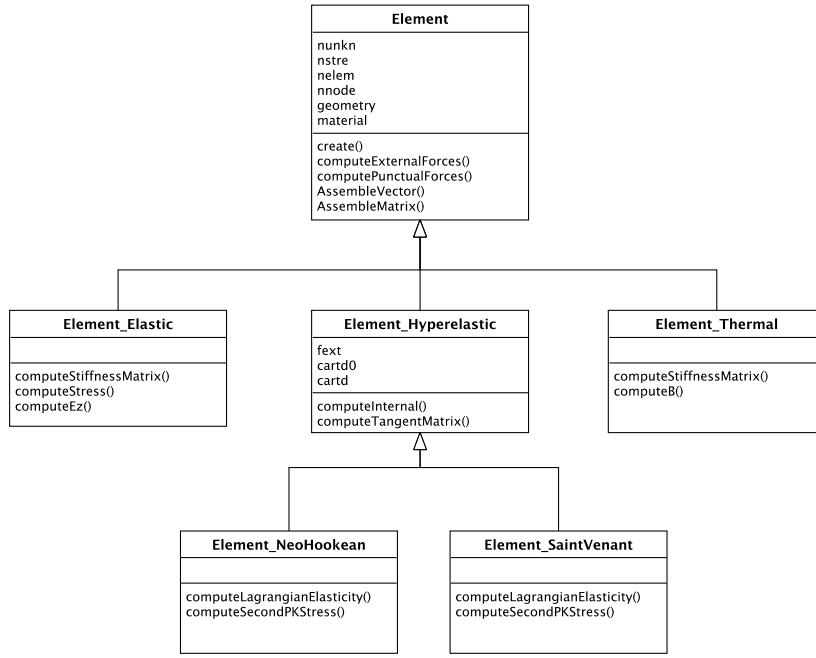
The last principle is *polymorphism* which designates the capability of object-oriented applications to interpret the same request differently depending on the object being processed [19]. Polymorphism is seen in Figure 3.3 inside the *Element* class. The method *AssembleVector()* or *AssembleMatrix()* does not depend on the element object. Therefore, polymorphism avoids to create and *AssembleMatrix()* function for each element (*AssembleMatrix\_Elastic*,...), and thus allows to reduce code duplicities.

```

% Load the results for 2-d and 3-d tests
tests_fem={'test2d_triangle';
'test2d_quad';
'test3d_hexahedra';
'test3d_tetrahedra';
'test2d_triangle_neo'};

```

**Listing 3.3:** Test script.



**Figure 3.3:** An example of inheritance; the *Element* class.

## 3.2. FEM-MAT-OO

*FEM-MAT-OO* is a collaborative and modular FEM software developed in Matlab. This object-oriented software is able to solve structural, thermal and fluid-related problems.

### 3.2.1. Code Instructions

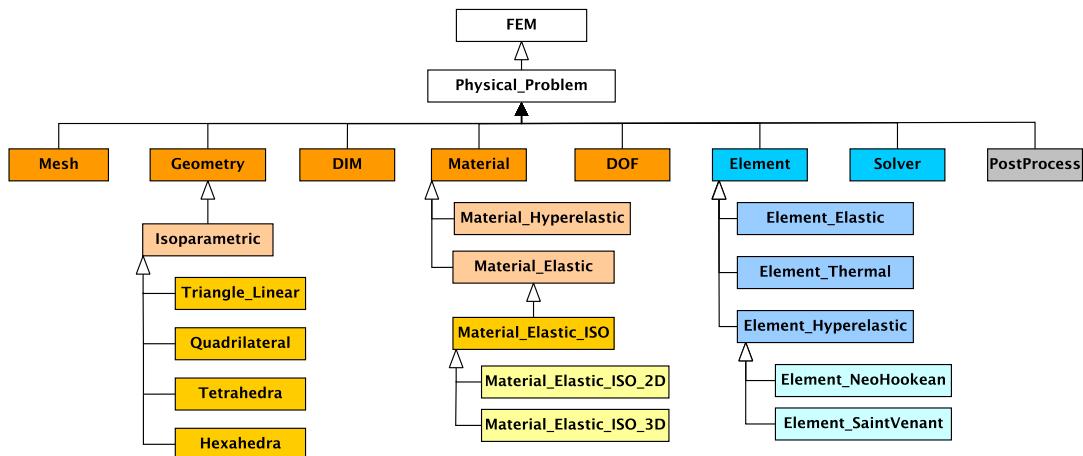
The correct steps for running *FEM-MAT-OO* are explained below.

1. **Create the mesh.** The CAD software used during the project is GiD. This software is both pre and post processor meaning that it allows to create the mesh and to visualize the results. After creating the geometry (2D or 3D), the boundary conditions will be applied. Eventually the element type is selected, and then the geometry is meshed. From this step we obtain a Matlab file with the corresponding coordinates, connectivities, Neumann and Dirichlet boundary conditions.
2. **Run FEM-MAT-OO.** The Matlab file is introduced in the *input* folder, inside *FEM-MAT-OO*. The code is run from *Main\_FEM.m* file, and it is mainly divided into three parts.
  - a) **Tests.** The pillar of test-driven development. Before executing the code it is necessary to ensure that everything is working properly. New tests can be added to the *test* folder.
  - b) **Preprocess.** In this part the input file is read, obtaining important variables such as the Cartesian derivatives or the Jacobian among others.

- c) **Computation.** Here the residual equation is computed until it accomplishes the required convergence criteria. Displacements, stresses and strains are calculated in this phase.
  - d) **Postprocess.** Eventually all the necessary result files are created here. These files are saved in the *output* folder.
3. **See the results.** Using GiD again we will be able to see the results by introducing the result files.

### 3.2.2. Class organization

The overall class organization of *FEM-MAT-OO* is depicted in Figure 3.4. The classes in orange and yellowish are related with the preprocess phase, while the blue classes have to do with the intermediate phase of computing. Eventually the postprocess phase takes place to obtain the result files.



**Figure 3.4:** *FEM-MAT-OO* UML diagram showing the main classes related to nonlinear FEM.

Mesh class is the responsible of reading all the *input.m* file, created previously with GiD. All the variables related to dimension, such as the dimension *ndim* itself or number of unknowns *nunkn*, are stored here. Geometry class is one the most important classes; the class where the basics of FEM are coded. Depending on the type of element, the most important variables are computed in the following table.

It is important to highlight that the attributes *cartd*, *dvolu* and *djacb* are computed for each gauss point. DOF class contains constrained and free degree-of-freedom indexes. Material class, as well as Element class, are the only classes which need of a *create* method in order to instantiate objects. In Material class the Lamé parameters  $\mu$  and  $\lambda$  are imposed.

Element class defines the generic behaviour of finite element. Basically both left and right-hand sides of  $K u = f$  are computed, to then obtain the value of  $u$ . The functioning of this class is explained in depth in Section 3.2.4. Solver class just permits to compute  $u = K^{-1}f$ . Finally, PostProcess class is in charge of creating the result files for a later visualization in GiD.

Class	Geometry
properties	% description
ngaus	Number of integration points
posgp	Coordinates of these integration points.
weigp	Weight of the integration points.
shape	Shape functions $N_i$ .
deriv	Partial derivatives of shape functions $\partial N_i / \partial \xi$ and $\partial N_i / \partial \eta$ .
cartd	Partial derivatives of shape functions w.r.t. Cartesian coordinates $\partial N_i / \partial X_i$ .
dvolu	Differential of volume $dv = J d\xi d\eta$ .
djacb	Jacobian, which is $\det \partial(x, y) / \partial(\xi, \eta)$ . Do not confuse with $J = \det F$ .

**Table 3.1:** Main properties of Geometry class.

### 3.2.3. Test-driven Development

Test-driven development has been implemented in *FEM-MAT-OO*. This development process relies on using tests to drive the design of software. Firstly the requirements are turned into specific test cases, and secondly the software is only improved whether it is able to pass the new tests.

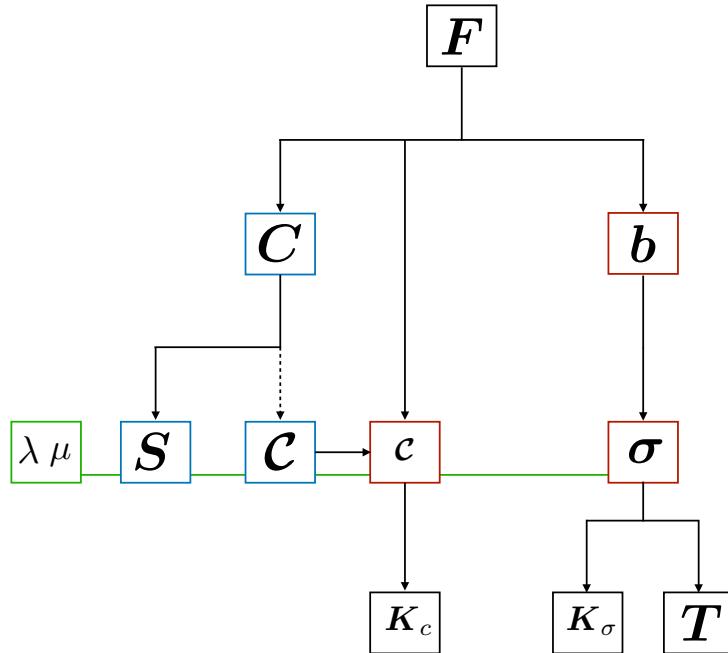
The motto of test-driven development is “*Red, Green, Refactor*”, where red and green means fail and pass respectively, and refactor is the process of restructuring the existing code. The sequence is the following:

1. **Add a test.** Every test is a requirement the code must accomplish.
2. **Run all tests.** Not only the new added test but also the complete test list must be passed. This step ensures the new test works. Figure 3.3 shows the tests needed to pass before executing the code.
3. **Write code.** New features are included in the code. At this stage the code needs to work, so aesthetics is forgotten.
4. **Validation.** If any problem arises it means the code meets the test requirements, and does not break or degrade any existing features. If the code does not pass the tests it will be adjusted until the requirements are met.
5. **Cleanup.** In this step the new code is cleaned up and it is usually redistributed in a logical way. This step is also known as code refactoring.

Developing a collaborative code implies an extra difficulty. Although each developer has to work in his project topic, the main structure of the code, or *kernel*, is developed jointly. Specially during the beginnings the test-driven development provides the developer with confidence to write and add your own classes or methods without interfering in someone else’s code. Owing to the fact that all the developers are not in the same place, coordination and resolution issues are mainly managed by the thesis advisor.

### 3.2.4. Hyperelasticity Core

The hyperelasticity core inside *FEM-MAT-OO* is depicted in Figure 3.5. As it is seen everything spins around the gradient tensor of deformations  $\mathbf{F}$ . Two different paths come from  $\mathbf{F}$ ; the left and right Cauchy-Green tensors. The blue and red rectangles framing the tensors indicate material and spatial configurations, respectively. Tensors  $\mathbf{C}$  and  $\mathbf{b}$  define both  $\mathbf{S}$  and  $\boldsymbol{\sigma}$  stress tensors in the corresponding configuration.



**Figure 3.5:** Diagram showing the dependencies of the most important quantities related to hyperelastic analysis.

The Lagrangian elasticity tensor  $\mathcal{C}$  is computed by using  $\mathbf{C}$ . In Figure 3.5 this relationship is expressed by using a discontinuous line since depending on the constitutive model  $\mathbf{C}$  is used or not. The calculation of the Eulerian elasticity tensor involves  $\mathcal{C}$ ,  $\mathbf{F}$  and  $J = \det \mathbf{F}$ .

Modularity is one of the most interesting features of *FEM-MAT-OO*. It allows for example to add new elements under the `Isoparametric` class, or introduce new definitions of elements. Modularity benefits not only extends to structure the code, but also allows to expand and introduce new implementations without the risk of damaging the existing code. An illustrative example of modularity is seen inside `Element_Hyperelastic`. For computing the stiffness tangent components  $\mathbf{K}_c$  and  $\mathbf{K}_\sigma$ , and the internal energy  $\mathbf{T}$  only the spatial tensors are needed. Therefore, instead of defining the material quantities ( $\mathbf{S}$ ,  $\mathcal{C}$ ), and the spatial quantities ( $\boldsymbol{\sigma}$ ,  $c$ ), the use of push-forward operations reduce to two the number of expressions which need to be defined per type of element.

Model 1	$\mathcal{C}_1$	$\mathbf{S}_1$
Model 2	$\mathcal{C}_2$	$\mathbf{S}_2$
⋮		

**Table 3.2:** `Element_Hyperelastic` child implementation.

### 3.2.5. Source Control

Source control, also known as version control, is a key component of software configuration management. It allows to manage efficiently large computer programs with different developers working at it at the same time.

The development of a software such as *FEM-MAT-OO* implies a large journey with multiple versions. At simplest level it could be possible to make copies of the different versions of the program, and label them appropriately. However, this method is inefficient as many similar copies should be maintained. In addition, this practice often leads to mistakes. Revision control track changes automatically so you can retrieve an older version if necessary. Moreover, version control allows the code to be edited by the team simultaneously.

#### 3.2.5.1. Git

Git<sup>2</sup> is an open source distributed version control system, and it is the one used in this work. The following points are the basis of Git.

- **Branches.** *FEM-MAT-OO* repository<sup>3</sup> has one branch named *master*, which is considered to be the definitive branch. By creating different branches it is possible to work on different versions of a repository at the same time.
- **Commit.** The key feature of Git. When a commit is done, all the changes are saved. Each commit has an associated message, which is a description explaining the change. This description is essential when we are forced to go back and use a previous version.
- **Merge.** After passing the tests, the new changes may merge with other branches. In the case we want to merge with the master a pull request must be proposed.

Figure 3.6 shows each of the branches as horizontal lines. Many branches diverge into separate paths creating, in turn, new branches. Each dot represents a commit, which is a *snapshot* of every point in the project's history. Branches free of errors and bugs are eventually merged into the master branch.



**Figure 3.6:** *FEM-MAT-OO* network graph. Master branch is represented with a black line.

<sup>2</sup>Sometimes Git is confused with *GitHub* which is the hosting service for Git repositories.

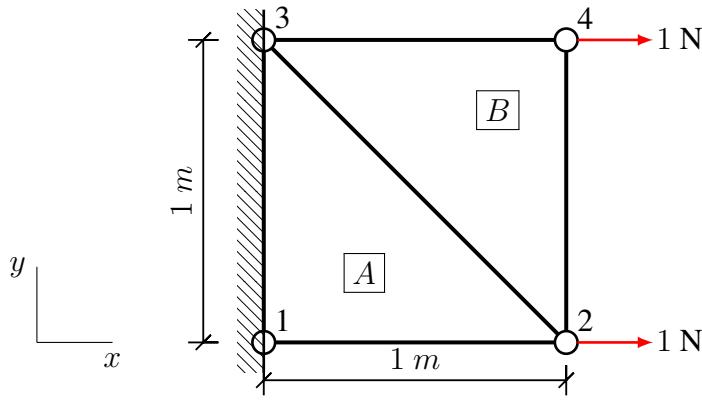
<sup>3</sup>*FEM-MAT-OO* repository <https://github.com/FerrerFerreAlex/FEM-MAT-OO.git>.

# CHAPTER 4. RESULTS

## 4.1. Test Problem

The test problem consists on a very basic geometry which helps to both develop the code and find errors easily. Once it is verified it will become a *test*. This implies that before modifying the code, first the test will need to be passed.

The structure showed in Figure 4.1 is composed by 2 linear triangle elements, *A* and *B*. Each node of the triangle has 2 DOF defined by the displacement components  $\{u_x, u_y\}$ . The displacement of nodes 1 and 3 is restricted in both DOF whilst nodes 2 and 4 have a force of 1N applied in the positive direction of  $x$  axis.



**Figure 4.1:** Simple two-dimensional example.

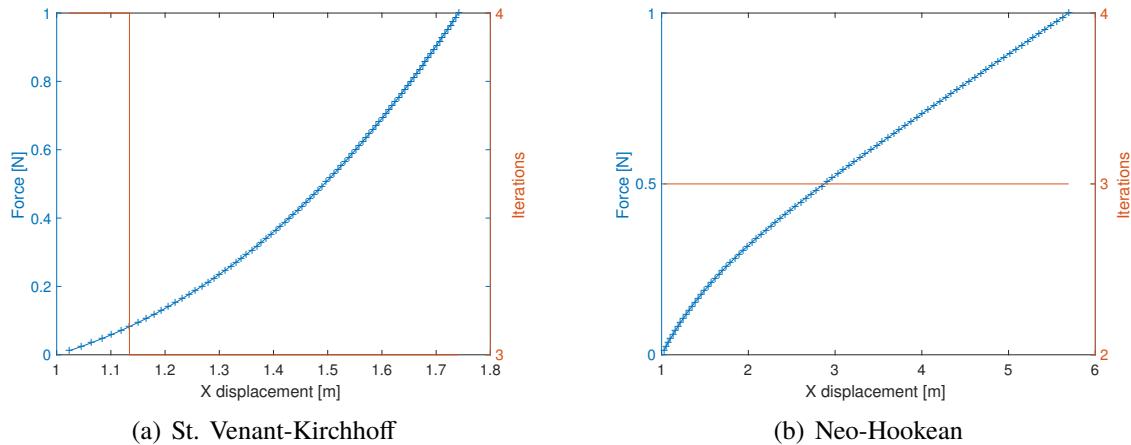
When solving the nonlinear problem, the solution at the beginning of the simulation will be very similar to the linear problem solution. Figure 4.2 shows for a given external nodal forces the x-coordinate position of node 4. The external force  $\mathbf{F}$  is applied in a series of increments, in this case 85.

Assuming plane strain deformation, this example is tested for both Saint Venant and Neo-Hookean models. It is seen that the Newton-Raphson method converges at each increment by doing 3 or 4 iterations per increment, evidencing quadratic convergence. The tolerance of each increment is  $10^{-12}$ . Node 4 elongates roughly a 80% for the Saint Venant case and a 600% for the Neo-Hookean case, and thus implies large deformation.

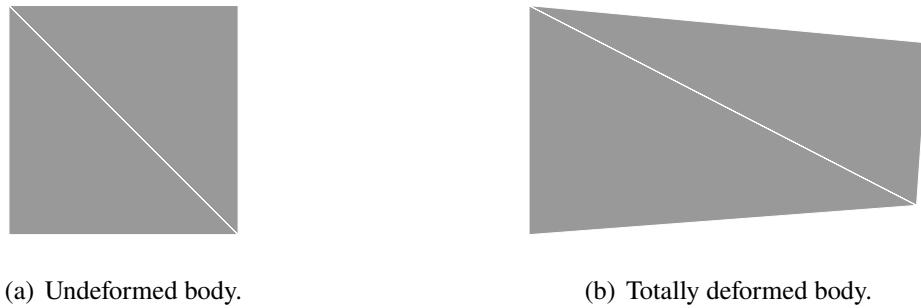
General	Test	Model	SaintVenant
cload	1	$\mu$	0.3446
nincr	5	$\lambda$	0.5661
nelem	2	type	triangle_linear

**Table 4.1:** Summarized parameters.

As it is seen in Figure 4.3 the deformation of the body is nearly symmetric. However, by focusing on the deformed body, node 4 is slightly more stretched than node 2. This can explained by looking how elements are restricted. Element *A* has two nodes attached, while element *B* only one.



**Figure 4.2:** Load-displacement curve of node 4. The external load is applied in a series of 85 increments.



**Figure 4.3:** Deformation of body.

## 4.2. Preliminary Cases

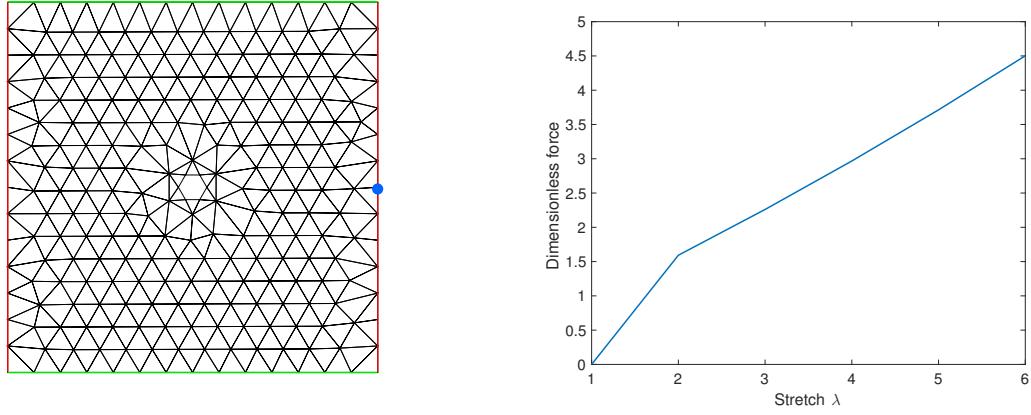
During the development of the code different examples have been created in order to introduce new functionalities and features. The most remarkable ones are shown in this section. The cases are separated depending on the dimension of the problem, 2D or 3D.

### 4.2.1. 2D Cases

#### 4.2.1.1. Strip With a Hole

In this example an initial  $6.5 \times 6.5 \text{ m}^2$  strip with a hole 0.5 m diameter is stretched in the horizontal direction and constrained in the vertical direction as shown in Figure 4.4(a), where both red sides are subjected to a tensile load. Dirichlet conditions are expressed in green. In blue the node 48, the one monitored. A plane strain Neo-Hookean material model is used.

As it is explained in Section 2.5, the Neo-Hookean model is typically used in rubber-like materials. It is the Saint Venant constitutive model the one employed to simulate metals, and thus it is the one selected for the rest of the models. Figure 4.5 shows how node 48 is stretched depending on the number of load increments `nincr`. Although the use of a



(a) Mesh &amp; boundary conditions of strip-hole case.

(b) Longitudinal stretch.

**Figure 4.4:** Force versus displacement of node 48.

higher number of increments reduce the number of iterations it is checked that without distinction of `nincr` the strip is stretched to six times its horizontal length, as [20] affirms.

General	Strip-hole	Model	NeoHookean
cload	1	$\mu$	0.3446
nincr	85	$\lambda$	0.5661
nelem	428	type	triangle_linear

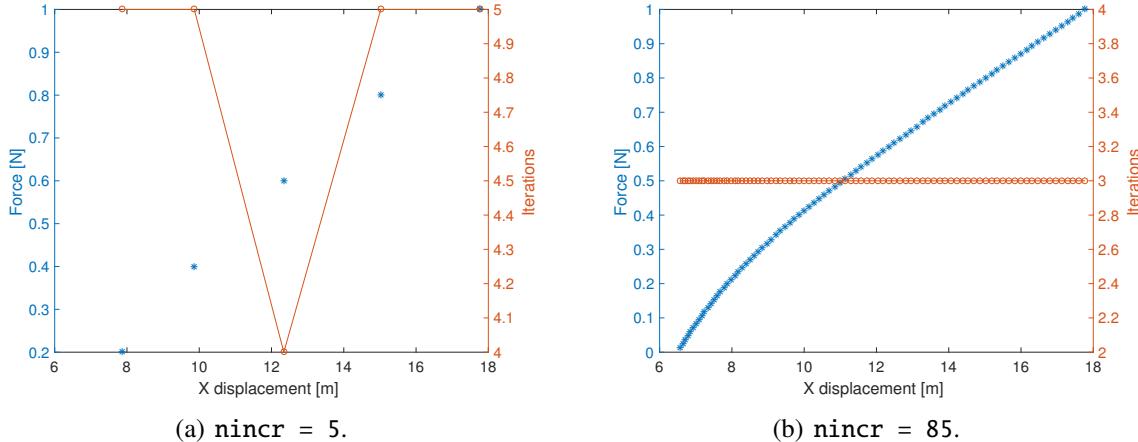
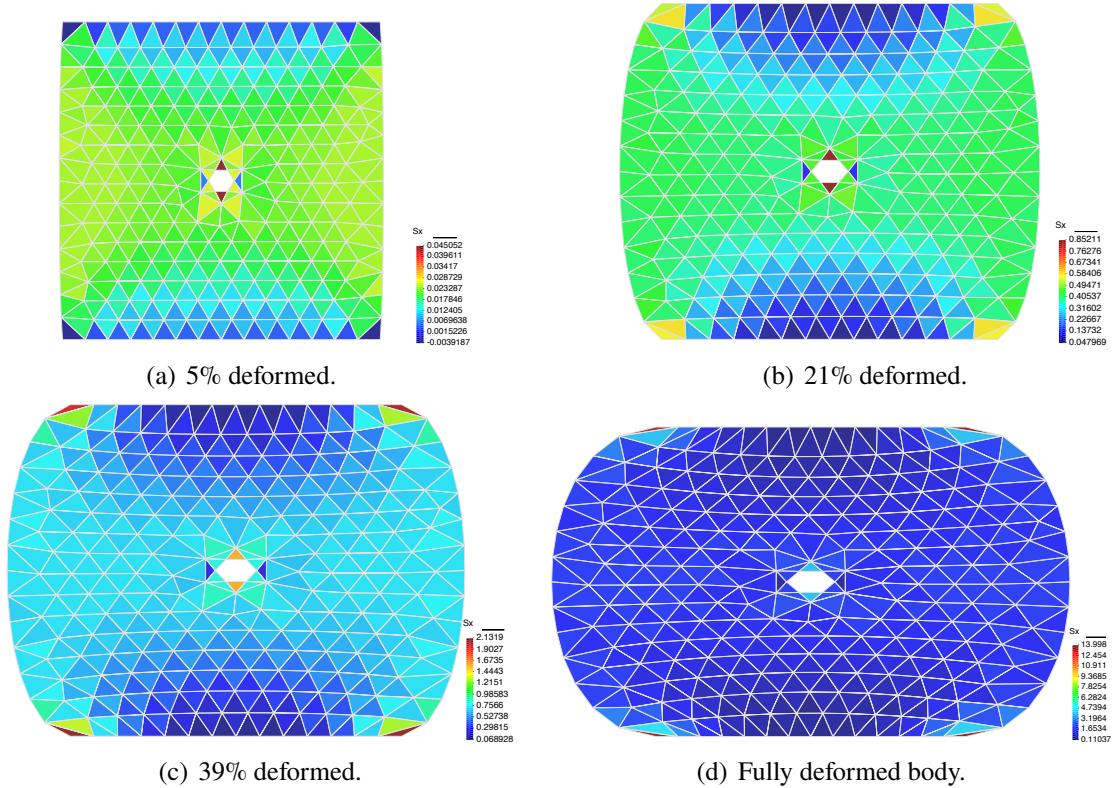
**Table 4.2:** Summarized parameters of the strip-hole case.(a)  $nincr = 5$ .(b)  $nincr = 85$ .**Figure 4.5:** Load-displacement curve of node 48.

Figure 4.6 shows both the deformation and the *real* or Cauchy stress in the  $x$  direction. In the first frame 4.6(a) it is seen how only the corners are subjected to compression stresses. Besides, the elements located at the top and bottom of the hole are enduring two times more stress than any other element. Corner elements suffer the highest distortion. Figures 4.6(c) and 4.6(d) show how their volume is been reduced.



**Figure 4.6:** Distribution of  $\sigma_x$  for different increments.

#### 4.2.1.2. Quadrilateral

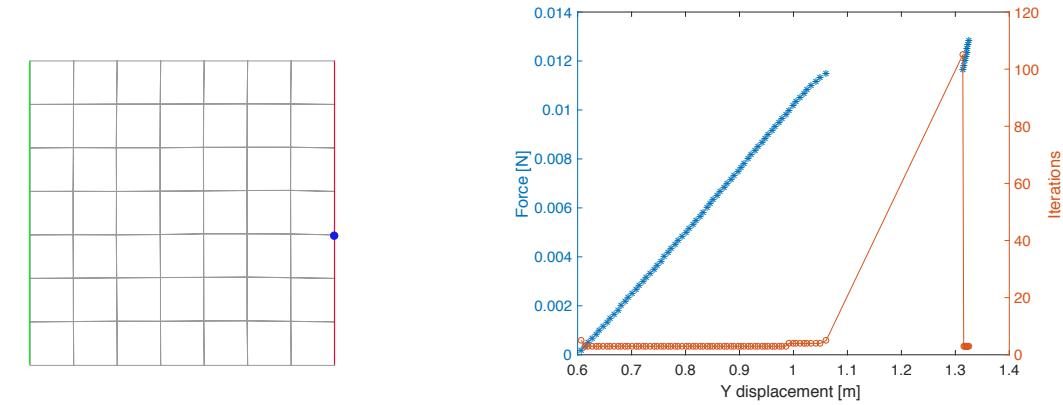
This 2D example introduces two new features. It uses quadrilateral elements, involving explicit *gauss integration*. Until this example all cases were using linear triangular elements with only 1 gauss point. Besides, in this case the load is applied normal to the horizontal axis introducing a bending moment in the structure.

General	Quadrilateral	Model	SaintVenant
cload	0.01	$\mu$	0.3446
nincr	2000	$\lambda$	0.5661
nelem	25	type	quadrilateral

**Table 4.3:** Summarized parameters of the quadrilateral case.

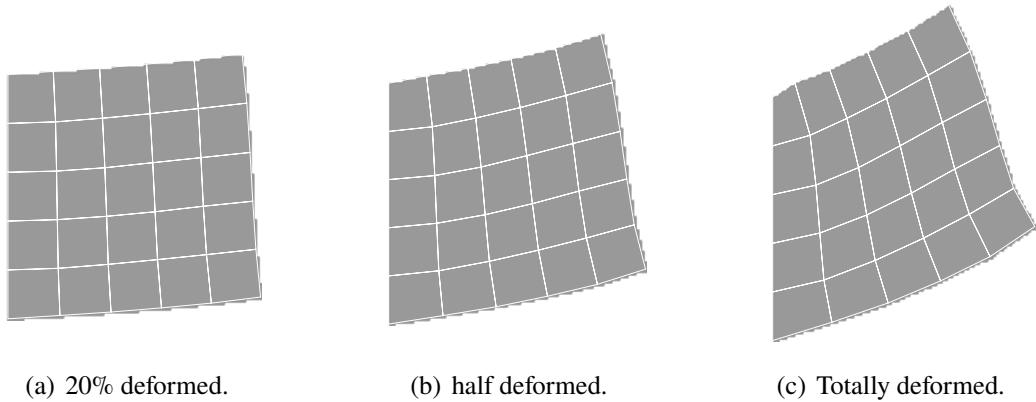
Figure 4.7(a) shows the mesh composed of 25 elements. Dirichlet boundary conditions are expressed in green, while Neumann conditions are coloured in red. In order to find a physical solution, the force is enforced to be 0.01 N. Figure 4.7(b) shows how a too large force can cause Newton-Raphson to find a non-realistic solution.

Figure 4.8 shows the deformation of the quadrilateral and how applying the same force in the same face, but now axially, leads to proper results for **nincr**= 50.



(a) Mesh &amp; boundary conditions.

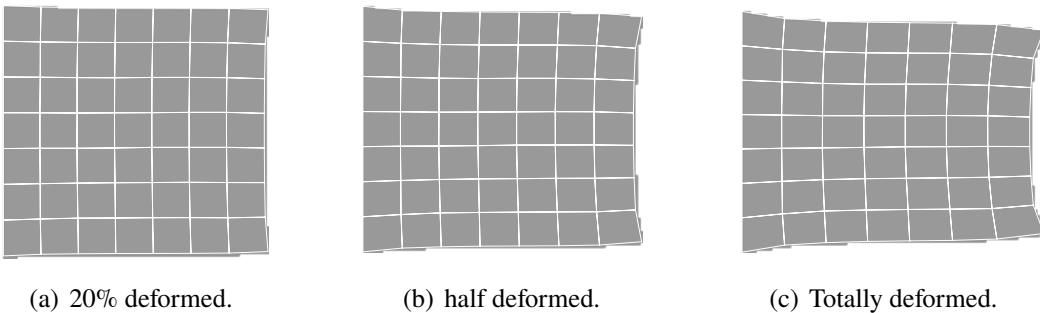
(b) Response diagram.

**Figure 4.7:** Mesh diagram and response diagram monitoring node 6.

(a) 20% deformed.

(b) half deformed.

(c) Totally deformed.

**Figure 4.8:** Deformation process of a quadrilateral subjected to a normal force.

(a) 20% deformed.

(b) half deformed.

(c) Totally deformed.

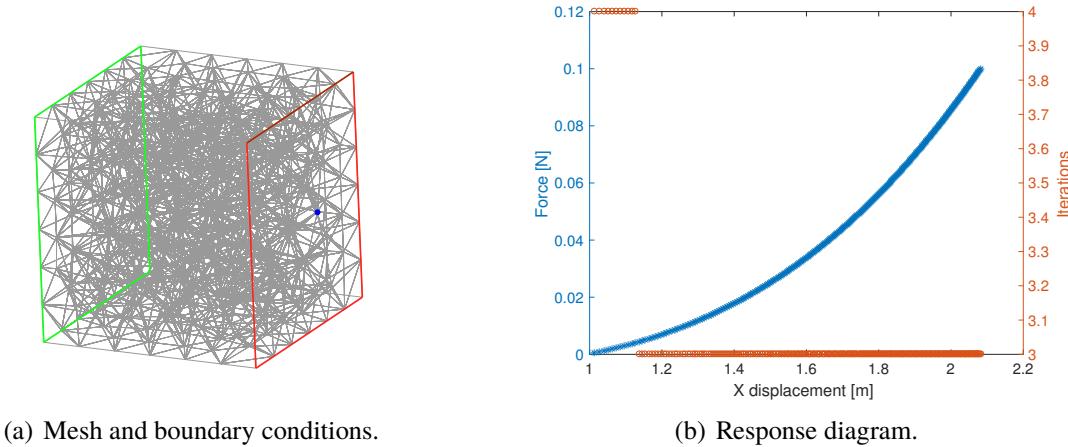
**Figure 4.9:** Deformation process of a quadrilateral subjected to an axial force.

## 4.2.2. 3D Cases

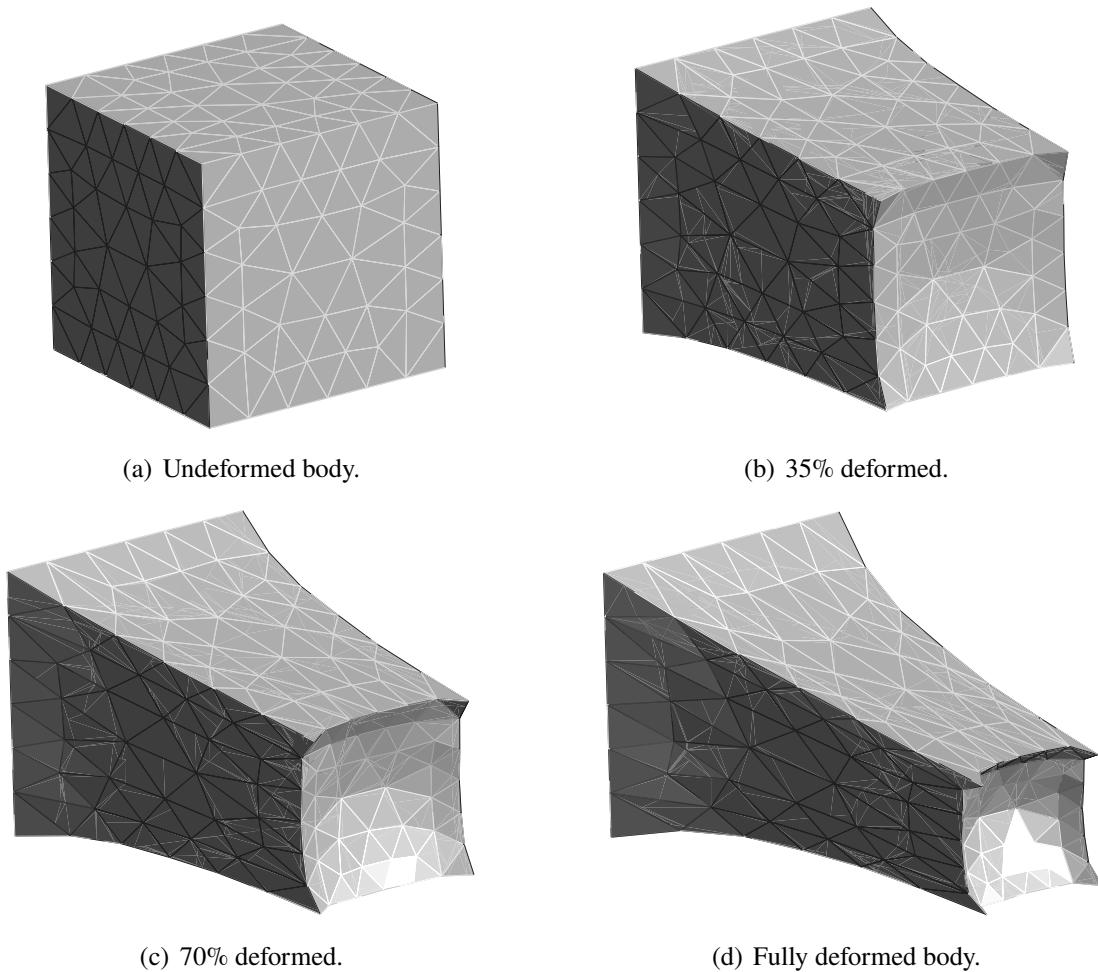
### 4.2.2.1. Hexahedron

An hexaedron of dimensions  $1 \times 1 \times 1 \text{ m}^3$  is the first 3D case addressed. The displacement on the left face (marked in green in Figure 4.10(a)) is constrained. In turn, the opposite face is subjected to a surface tension load. From Figure 4.10(b) it is seen that the force-

displacement response is continuous and growing curve without any limit point, as opposed to Figure 4.7(b).



**Figure 4.10:** Mesh and boundary conditions of Hexahedron I case. The monitored node 342 is coloured in blue.



**Figure 4.11:** Hexaedron deformation process.

The hexaedron element is deformed more than a 200%. It seems to be suffering an *extrusion* process. Thus, depending on the constitutive model a metal or a polymer could be simulated.

<b>General</b>	<i>Hexaedron</i>	<b>Model</b>	SaintVenant
cload	0.1	$\mu$	0.3446
nincr	285	$\lambda$	0.5661
nelem	1743	type	tetraedra

**Table 4.4:** Summarized parameters of the hexaedron case.

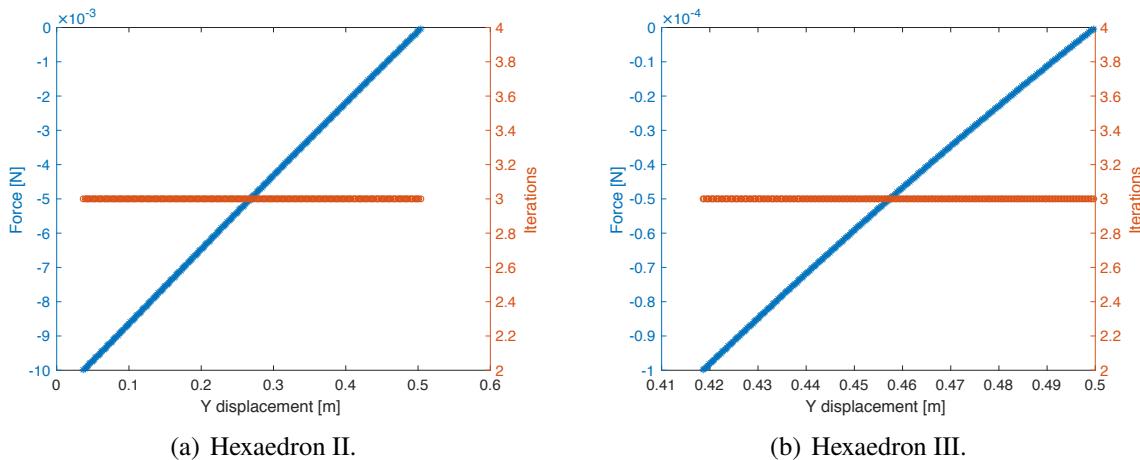
#### 4.2.2.2. Hexahedron II and III

Before introducing slender structures it is necessary to check how a 3D structure behaves under normal loads. In this problem, the same hexaedron case is used, now subjected to a normal load. Here, it is mandatory to reduce the load step size to achieve convergence.

<b>General</b>	<i>Hexaedron II</i>	<b>Model</b>	SaintVenant
cload	0.01	$\mu$	0.3446
nincr	285	$\lambda$	0.5661
nelem	88	type	tetraedra

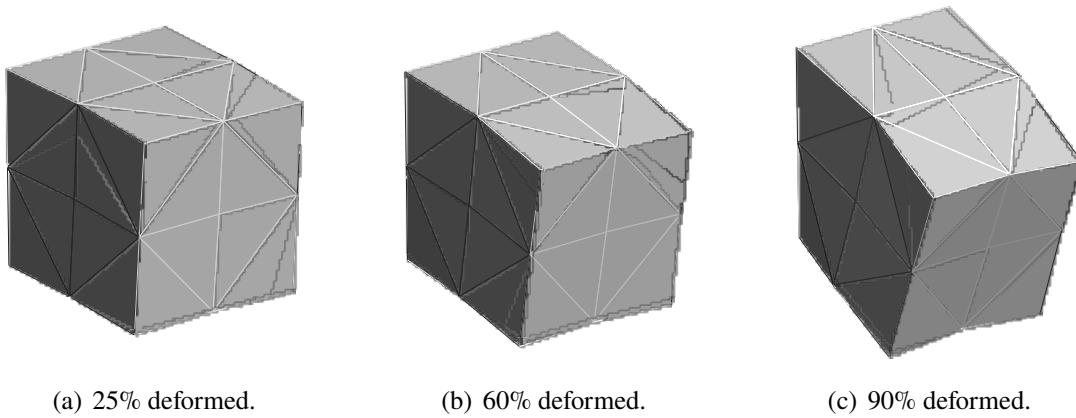
**Table 4.5:** Summarized parameters of the hexaedron II case.

From this hexaedron case we observed that large normal forces lead to non-physical results, where elements have to perform really large deformations, sometimes folding over itself leading to a negative value of  $J = dv/dV$ . Therefore, for the current Lamé parameters, the force is reduced to find realistic results.



**Figure 4.12:** Response diagrams of hexaedron cases II and III.

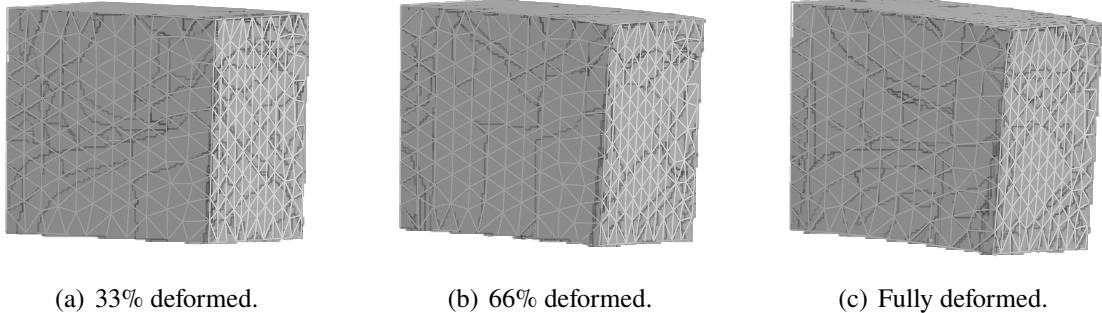
The nodal force value is reduced consistently with the mesh size in order to keep the resultant force constant. The node monitored, which is located in the face where the force is applied, moves a 20%.



**Figure 4.13:** Deformation process of a hexaedron subjected to normal forces.

General	Hexaedron III	Model	SaintVenant
cload	0.0001	$\mu$	0.3446
nincr	285	$\lambda$	0.5661
nelem	9742	type	tetraedra

**Table 4.6:** Summarized parameters of the hexaedron III case.



**Figure 4.14:** Deformation process of a hexaedron subjected to normal forces.

#### 4.2.2.3. Slender Structure

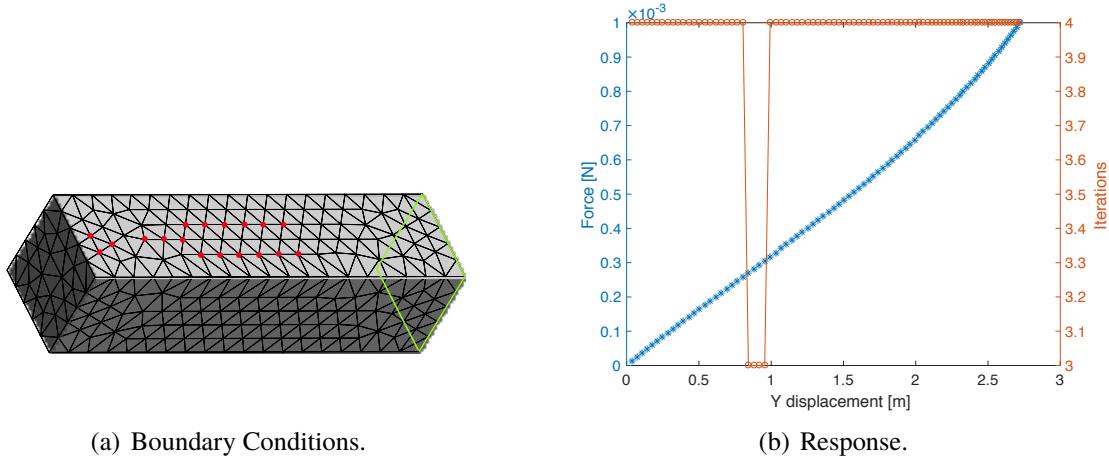
The slender structure example represents the last step before addressing the wing final test. This beam, which is called *slender* to differentiate it from hexaedral cases, is four times larger than the hexaedron. Figure shows the boundary conditions. In this case the load is not applied in the opposite face, but in the upper face. This load is distributed among 18 nodes (Figure 4.15(a)) obtaining a reasonable force to do not fail in convergence as in previous example.

solving the convergence problem caused in the latter cases.

Figure 4.15(b) shows how almost all the force increments converge within four iterations. The beam is deflected 2.6 m, and thus the nonlinear response is noticeable.

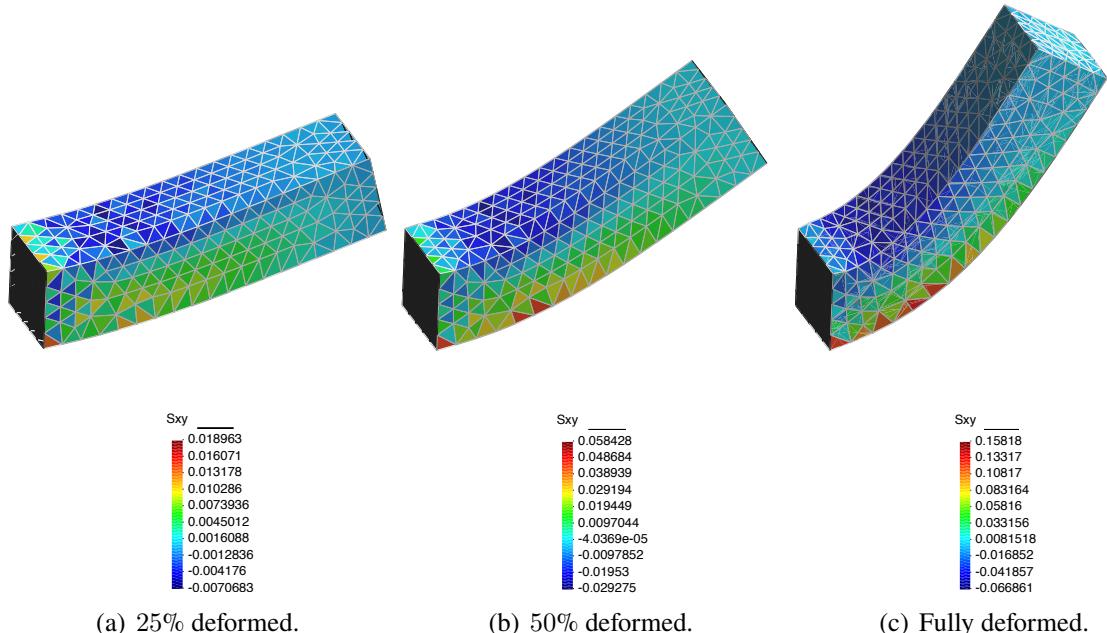
Figures 4.16 and 4.17 show  $\sigma_{xy}$  stress-component distribution. It is seen how compression and tension stresses affect both upper and bottom faces respectively.

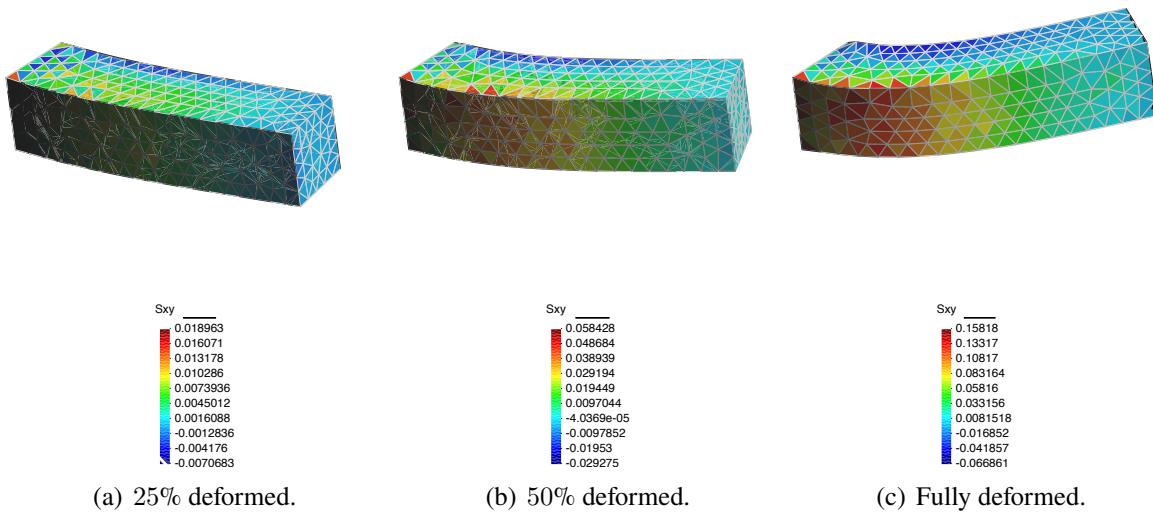
General	Slender	Model	SaintVenant
cload	0.001	$\mu$	0.3446
nincr	85	$\lambda$	0.5661
nelem	4315	type	tetraedra

**Table 4.7:** Summarized parameters of the hexaedron case.

(a) Boundary Conditions.

(b) Response.

**Figure 4.15:** Boundary conditions and response diagram of slender case. (a) In red the 18 nodes where the force is applied. Fixed displacements are coloured in green. (b) The monitored node is located in the tip of the beam.**Figure 4.16:**  $\sigma_{xy}$  stress-component distribution for the slender beam case.



**Figure 4.17:**  $\sigma_{xy}$  stress-component distribution showing the lower part of the beam.

## 4.3. Wing Loading Test

The results obtained in the previous cases provides us with the necessary tools to face the wing loading test. The development of the test could be summarized in three topics.

- **Design of the test.** A comprehension of the wing bending test is necessary to design the process. Some questions may arise when facing the problem: how and where are the loads applied? How large are the expected deflections? Is there any benefit in simulating the test?
- **Design of the wing.** The geometry of the wing is based on a real wing, introducing several simplifications. To complete the structural design it is necessary to select the proper constitutive model. In addition, the material parameters have to be chosen according to the wing material.
- **Simulation of the test.** The parameters of the forces and conditions have been defined to obtain a realistic solution.

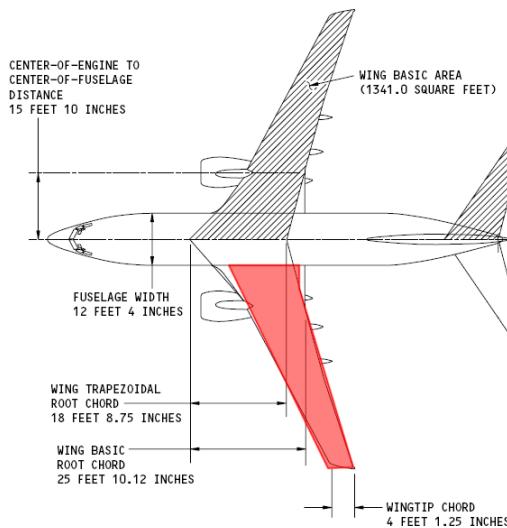
### 4.3.1. Regulations & Procedure

The wing loading test is an important aspect to consider by manufacturers. To be certified by the Federal Aviation Administration (FAA), all airliners must withstand 150 percent of the most extreme forces the airplane is ever expected to experience while in service [21]. These extreme forces are typically encountered during a gust response.

Due to the research nature of the load tests, it is imperative that the loads be applied at small increments so the results of progressively increased loads could be documented. Also, in the case of test failure, a single application of the testing load could cause catastrophic failure without any forewarning. On the other hand, an incremental application of the load may show lesser signs of failure before the aircraft would break beyond repair [22]. Nevertheless, in this case we are forced to apply the force in increments as it is explained in Section 2.7.4.1.

### 4.3.2. Structural Design

The wing structure is based on the *Boeing 737* airfoil, however several simplifications are done. It is designed omitting the movable parts, such as high-lift devices, and wingtip devices. Besides, the wing is divided into three main parts: root, mid and tip, each part with its corresponding airfoil.



**Figure 4.18:** Boeing 737-800 plan view schematic. In red the designed area.

Table 4.8 shows the relationship between the model and Boeing data. Most of the parameters are really similar. For instance, the real span is larger due to the wingtip device. The Lamé parameters selected correspond to aluminium, and as it is said previously Saint Venant is the proper model to simulate metals.

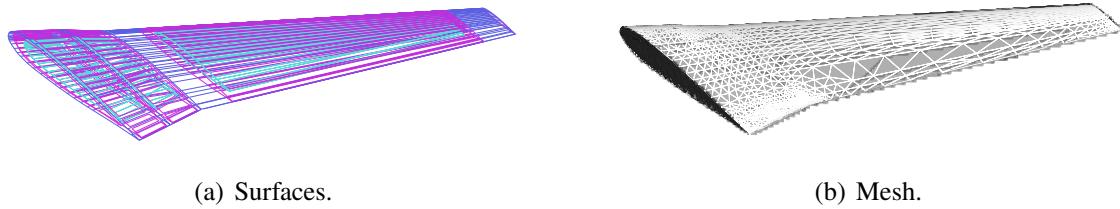
Definition	Model	737-800
Span /2	14.64 m	15.28 m
Root chord	7.32 m	7.76 m
Wingtip chord	0.805 m	0.782 m
Sweep	29.6°	25°
Wing area /2	56 m <sup>2</sup>	62.43 m <sup>2</sup>

**Table 4.8:** Comparison between model and Boeing 737-800 data.

The airfoils are obtained using an [online database](#), which provides the nodal coordinates. These coordinates are then represented in GiD. The procedure consists of connecting the airfoils by using lines, creating the surfaces, and eventually creating the volumes (see Figure 4.19).

### 4.3.3. Meshing

The volumes obtained (Figure 4.19(a)) are meshed by using tetrahedra elements, being 17876 the total number of elements. Tetrahedra elements allow a more efficient meshing process. Moreover these elements are recommended because of the ease to adapt to any



**Figure 4.19:** Wing design and meshing process. (a) Process of creating structure surfaces and volumes, in pink and light blue respectively. The three different airfoils can be noticed. (b) Mesh composed by tetrahedral elements.

General	Wing	Model	SaintVenant
cload	1750	$\mu$	$26315 \times 10^6$
nincr	40	$\lambda$	$51084 \times 10^6$
nelem	17876	type	tetraedra

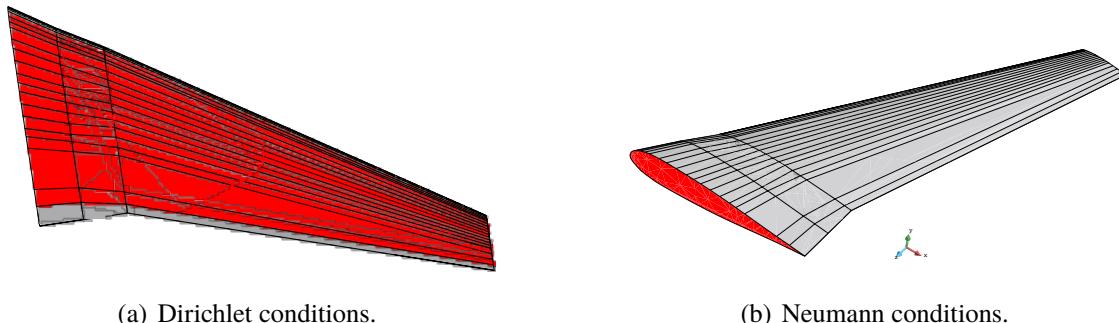
**Table 4.9:** Summarized parameters of the wing case.

geometry. This mesh is solid, meaning that it is filled up with elements. This is not realistic at all, however it is a good approximation. Future implementations would improve wing structure by introducing shells and a complete wing-box design.

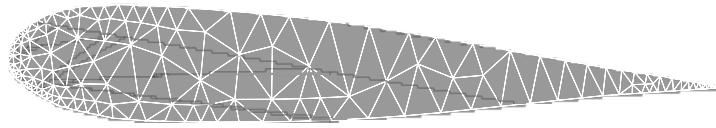
In order to reduce the number of elements the model is scaled by a factor of 7.32, so that the current *semispan* is 2 m.

#### 4.3.4. Boundary Conditions

The imposed forces are assumed to be in the positive direction of  $y$  axis (see Figure 4.20), normal to the reference configuration. This assumption causes larger stresses in the joining between the wing and the fuselage. The punctual forces are distributed along the extrados as it is seen in Figure 4.20(a). Owing to the fact that forces are punctual, some slim or low-volume elements have to endure large stresses. These elements are mainly located in the leading edge. In turn, displacements are restricted in the wing-body join.



**Figure 4.20:** Wing boundary conditions.



**Figure 4.21:** Root airfoil. Each airfoil is composed by 45 nodes. Maximum thickness is 14 cm.

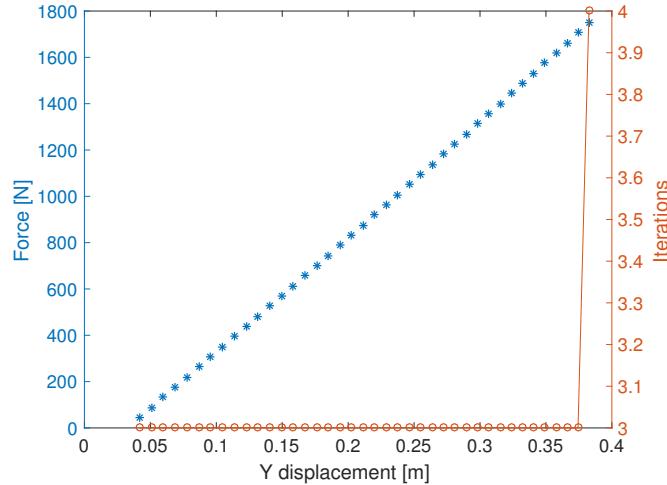


**Figure 4.22:** Superimposed deflection of the wing.

### 4.3.5. Simulation Results

The total force applied is  $940 \times 1750 = 1645$  kN. This value is obtained to be in physical correspondence with the other selected parameters. The response is depicted in Figure 4.23. The monitored node is located in the middle of the wingtip, and it undergoes a deflection of 34 cm, which is a 17%.

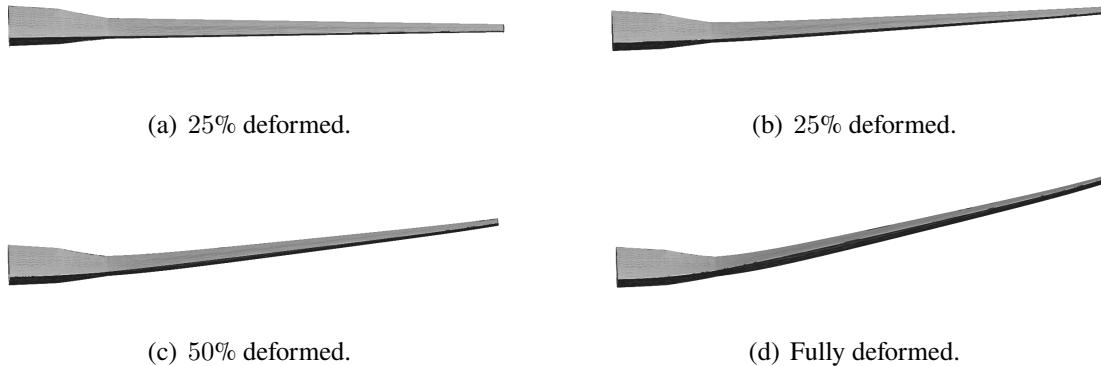
By taking the example of the wing-up bending test of Boeing 777, we know the maximum deflection achieved was 4 m (or 22%). This test, which happened in 1995, serve us as reference since Boeing 777 wing did not implement such advanced composite materials.



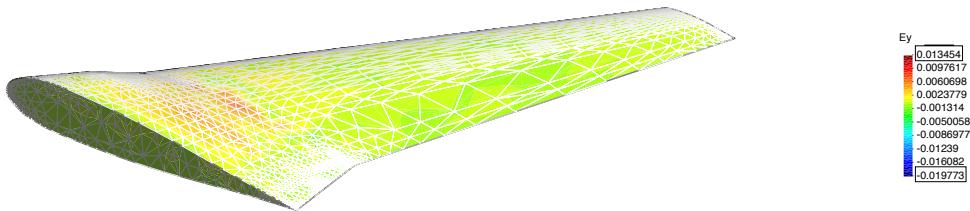
**Figure 4.23:** Response diagram of the wing.

The wingtip deflection in Figure 4.24(d) is 34 cm, creating a bank angle  $\phi \approx 10^\circ$ . If

this example was simulated without taking into account geometric nonlinearity, a body rotation of  $2.5^\circ$  would induce a non-desirable *fictitious* strain of the order of  $10^{-3}$ . As we can observe, in Figure 4.25 most of the wing undergoes strains of the same order of magnitude. The  $e_y$  strain of the wing-fuselage join is positive due to the punctual forces are completely normal to the wing forcing the structure to work under tension.

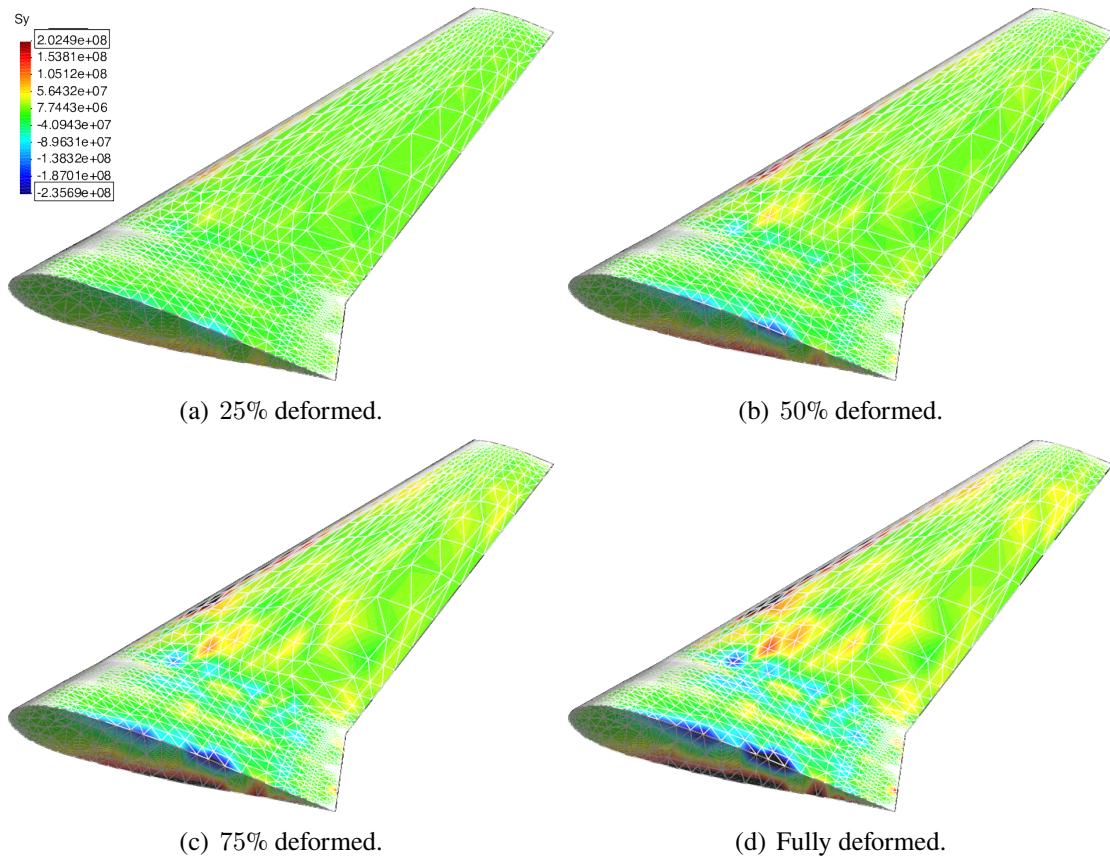


**Figure 4.24:** Deformation of the wing.

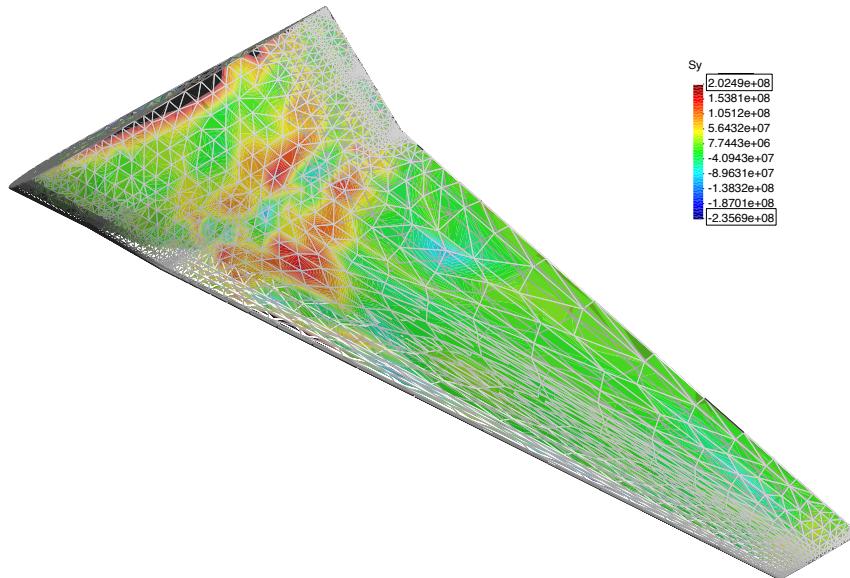


**Figure 4.25:** Final  $e_y$  strain-component distribution at the extrados.

The evolution of the  $\sigma_y$  stress-component is illustrated in Figure 4.26. The stress is scaled as to see clearly the zones with larger stresses. The compression and tension stress are observable in the upper and lower parts of the root, respectively. Most of the zones located in the intrados, as it is seen in Figure 4.27, are under tension stress. The zones in yellow are within the yield strength of aluminium alloy 6061, which is about 55 MPa. Nevertheless, in the orange and red zones this yield strength is exceeded. This limit, as well as the tensile strength, should be taken into consideration in the case of a real design.



**Figure 4.26:** Evolution  $\sigma_y$  stress-component distribution at the extrados.



**Figure 4.27:** Final  $\sigma_y$  stress-component distribution at the intrados.



# CONCLUSIONS

## Achievements

The aim of this work was to develop a collaborative software to solve generally nonlinear structural problems, being the wing bending test as the main motivation. This objective has been achieved by means of the Finite Element Method in conjunction of the nonlinear continuum analysis.

The journey begin with the mathematical foundation of FEM, introduced in Chapter 2. The complete method is expounded in Annexes. Due to the collaborative nature of this project, where different people are aiming to study a particular physical case, the object-oriented programming appears as the precise paradigm to create a FEM software. The basis of object-oriented programming, explained in Chapter 3, allow us to implement a first linear elasticity code able to solve simple examples. From this point, using the test-driven development, the code would begin to grow concurrently following different branches.

Nonlinearity, the main topic addressed in this work, is explained along Chapter 2 focusing on the most important matters such as kinematics and the concept of deformation gradient tensor, stress definitions and equilibrium equations, and constitutive relations where hyperelasticity arises as the suitable material model. Both Saint Venant and Neo-Hookean hyperelastic materials are implemented in *FEM-MAT-OO*, being the former the chosen for the wing test due to its ability to simulate a metal behaviour. At the end of the Chapter 2 the equilibrium equations are linearized before the discretization. Eventually, the Newton-Raphson and arc-length numerical procedures are prepared for a future computer implementation. The main structure and functioning of *FEM-MAT-OO* is discussed in Chapter 3, emphasizing the hyperelastic-related classes and functions.

The results are presented in Chapter 4. This phase begins with the achievement of quadratic convergence in a basic-geometry problem, and the definition of the *test* which becomes the most reliable piece of software. Before introducing any update, the test must be passed to make sure *FEM-MAT-OO* is working correctly. Problems of increasing geometry complexity are explained, as well as the needed conditions to achieve convergence. An aircraft wing, based on the Boeing 737, is modeled using the Saint Venant hyperelastic model. After the simulation a deflection within the range of real cases is accomplished.

*FEM-MAT-OO* framework is set in such a manner that permits to continue improving the software, not only in a more efficient way, but also introducing new features and capabilities. Referring to the nonlinear branch, more realistic results could be achieved by using shells which are more appropriate to the wing case. Moreover, an structure formed by more than one material, as well as considering plasticity behaviour could improve the simulation.

## Personal Growth

This work has supposed a huge challenge in many ways. Firstly, long parts of the project involved theoretical work such as the basis of Finite Element method or the nonlinear

mechanics. This sometimes can be frustrating because there is no *visible* progress. The most remarkable learning from this theoretical part are the vector and tensor manipulation, and the understanding of Finite Element method.

Secondly, and opposed to this theoretical part emerges the computer implementation. The nature of FEM provides object-oriented programming as the best way to develop and implement all FEM equations. However, it is the collaborative constraint the decisive reason to use the object-oriented paradigm. At first, object-oriented programming suppose can be seen as an obstacle, but once it is overtaken object-oriented programming improves considerably the rapidity of implementation.

Although several concepts and tools have been learnt, there are two which should be highlighted. Vectorization together with indicial notation, which serves as a half-way point between theoretical expressions and computer implementation, are the responsible of reducing the computational time. The second tool is *Git* a version control system which allows to separate the work independently in branches.

Lastly, I had to learn to work at distance during a period with all that implies. However, by setting small milestones and encapsulating any theoretical or software related issue we finally achieved the proposed objective.

# BIBLIOGRAPHY

- [1] P.J. Westwick. *Blue Sky Metropolis*. Western Histories. University of California Press, 2012. URL: <https://books.google.es/books?id=05IkDQAAQBAJ>. 1
- [2] The Boeing Company. *Boeing 787 Wing Flex Test*. 2010. URL: [https://www.wired.com/images\\_blogs/autopia/2010/03/index1.jpg](https://www.wired.com/images_blogs/autopia/2010/03/index1.jpg). 2
- [3] Paul N Patrone, Samuel Tucker, and Andrew Dienstfrey. Estimating yield-strain via deformation-recovery simulations. *Polymer*, 116:295–303, 2017. 2
- [4] Carlos A Felippa. Introduction to finite element methods. chapter 1, pages 3–6. 2001. 4
- [5] Carlos A Felippa. Introduction to finite element methods. chapter 6, page 4. 2001. 5
- [6] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 1. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 8, 9, 10, 11, 12, 13, 14, 65
- [7] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 4. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 17, 18, 25, 29, 33
- [8] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 5. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 35, 36, 41
- [9] COMSOL INC. Stress measures, 2018. URL: <https://www.comsol.com/blogs/why-all-these-stresses-and-strains/> [cited 6/03/2018]. 43, 44
- [10] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 6. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 44, 48
- [11] Ivan D Breslavsky, Marco Amabili, and Mathias Legrand. Nonlinear vibrations of thin hyperelastic plates. *Journal of Sound and Vibration*, 333(19):4668–4681, 2014. 48
- [12] Siamak Niroomandi, Iciar Alfaro, Elias Cueto, and Francisco Chinesta. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 81(9):1180–1206, 2010. 49
- [13] C.W. Macosko. *Rheology: principles, measurements, and applications*. Advances in interfacial engineering series. VCH, 1994. URL: <https://books.google.es/books?id=XXspAQAAQAAJ>. 49
- [14] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 8. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 52, 54, 55

- [15] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 9. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 65, 67
- [16] Bashir-Ahmed Memon et al. Arc-length technique for nonlinear finite element analysis. *Journal of Zhejiang University-SCIENCE A*, 5(5):618–628, 2004. 67
- [17] RI Mackie. Object oriented programming of the finite element method. *International Journal for Numerical Methods in Engineering*, 35(2):425–436, 1992. 69
- [18] Jamie Winter and Abhishek Gupta. Introduction to object-oriented programming in matlab. 69
- [19] V Kromer, F Dufossé, and M Gueury. An object-oriented design of a finite element code: application to multibody systems analysis. *Advances in Engineering Software*, 35(5):273–287, 2004. 71
- [20] J. Bonet and R.D. Wood. Nonlinear continuum mechanics for finite element analysis. chapter 10. Cambridge University Press, 1997. URL: <https://books.google.es/books?id=0RmLdrq1fI8C>. 79
- [21] Federal Aviation Regulations Part. 25: Airworthiness standards: Transport category airplanes. *Federal Aviation Administration, Washington, DC*, 7, 2002. 86
- [22] Jason Robinson. Structural testing and analysis of a joined wing technology demonstrator. Technical report, AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH STRUCTURAL DESIGN AND DEVELOPMENT BRANCH, 2004. 86
- [23] Carlos A Felippa. Introduction to finite element methods. chapter 11, pages 6,9. 2001. 100, 101, 103, 107
- [24] Carlos A Felippa. Introduction to finite element methods. chapter 14, pages 3–9. 2001. 109, 110, 112, 113
- [25] Carlos A Felippa. Introduction to finite element methods. chapter 14, pages 3–6. 2001. 115, 118
- [26] Carlos A Felippa. Introduction to finite element methods. chapter 16, pages 4–8. 2001. 120, 121, 122
- [27] Carlos A Felippa. Introduction to finite element methods. chapter 17, pages 1–5. 2001. 125, 127

## **APPENDICES**



# APPENDIX A. MATHEMATICAL FORMULATION OF FINITE ELEMENTS

In this appendix, we will briefly introduce the Finite Element method for linear elasticity.

## A.1. Bar Element

The one-dimensional bar element is selected since it is the simplest one. In the following sections more complex elements will be discussed. Several ways allow to explain the Finite Element method, however the Variational formulation and Galerkin method are addressed in this work.

### A.1.1. Variational Formulation

To illustrate the variational formulation, the finite element equations of the bar element will be derived from the Minimum Energy principle. The mathematical model of the bar assumes a linear elastic behaviour obeying Hooke's law, and that displacements and strains are infinitesimal.

#### A.1.1.1. Total Potential Energy Functional

In Mechanics of Materials it is shown that the internal energy density at a point of linear-elastic material subjected to a one-dimensional state of stress  $\sigma$  and strain  $\varepsilon$  is  $\mathcal{U} = \frac{1}{2}\sigma(x)\varepsilon(x)$ , where  $\sigma$  is to be regarded as linked to the displacement  $u$  through Hooke's law  $\sigma = E\varepsilon$  and the strain-displacement relation  $\varepsilon = u' = du/dx$ . The integration of the internal energy density over the volume of the bar gives the total internal energy

$$U = \frac{1}{2} \int_V \sigma \varepsilon dV = \frac{1}{2} \int_0^L F \varepsilon dx = \frac{1}{2} \int_0^L (E A u') u' dx = \frac{1}{2} \int_0^L u' E A u' dx. \quad (\text{A.1})$$

The external work potential  $W$ , which is the work performed by applied mechanical loads working on the bar displacements, collects contributions from two sources:

1. The distributed load  $q(x)$ . This contributes a cross-section density of  $q(x)u(x)$  because  $q$  is assumed to be already integrated over the section.
2. Any specified point load(s)  $P$ .

The second source can be folded into the first by writing any  $P$  acting at a cross section  $x = a$  as a contribution  $P\delta(a)$  to  $q(x)$ , in which  $\delta(a)$  denotes the one-dimensional Dirac delta function at  $x = a$ . Thus, the external energy can be expressed as

$$W = \int_0^L q u dx. \quad (\text{A.2})$$

The total potential energy of the bar is given by

$$\Pi = U - W, \quad (\text{A.3})$$

where  $\Pi$  the Total Potential Energy functional (TPE) and it only depends on the axial displacement  $u(x)$ <sup>1</sup>. By substituting the internal energy and external potential, Equation A.3 leads to

$$\Pi = \frac{1}{2} \int_0^L u' E A u' dx - \int_0^L q u dx. \quad (\text{A.4})$$

Let  $F = \frac{1}{2} u' E A u' - q u$  denote the integrand of  $\Pi$ . The Euler-Lagrange equation for  $\Pi$  is written as

$$\mathcal{L} = \frac{\partial F}{\partial u} - \frac{d}{dx} \frac{\partial F}{\partial u'} = -q - (E A u')', \quad (\text{A.5})$$

in which  $u$  and  $u'$  are taken as independent variable in the differentiation process. The stationary condition for  $\Pi$  is  $\mathcal{L} = 0$ , or

$$(E A u')u' + q = 0. \quad (\text{A.6})$$

This is the strong equation of equilibrium in terms of the axial displacement. Whether assuming  $E A$  constant, the strong equation is reduced to  $E A u'' + q = 0$ . This equation is not explicitly used in the FEM development, and it is instead replaced by  $\delta\Pi = 0$ , with the variation restricted over the finite element interpolation functions [23].

#### A.1.1.2. Principle of Virtual Work & Admissible Variations

The concept of admissible variation states “only the primary variable(s) of a functional may be varied”. For the TPE functional A.3 this is  $u(x)$ . Suppose that  $u(x)$  is changed to  $u(x) + \delta u(x)$ . A displacement variation  $\delta u(x)$  is said to be admissible when both  $u(x)$  and  $u(x) + \delta u(x)$  are kinematically admissible in the sense of the Principle of Virtual Work (PVW). A kinematically admissible axial displacement  $u(x)$  obeys two conditions:

1. It is continuous over the bar length, that is,  $u(x) \in C^0$  in  $x \in [0, L]$
2. It satisfies exactly any displacement boundary condition, such as the fixed-end specification  $u(0) = 0$  (see Figure A.1).

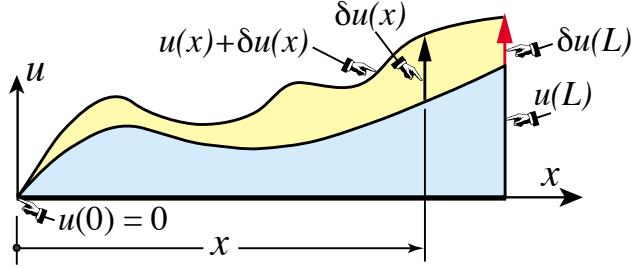
#### A.1.1.3. Minimum Total Potential Energy Principle

The Minimum Total Potential Energy (MTPE) principle states that the actual displacement solution  $u^*(x)$  that satisfies the governing equations is that which results  $\Pi$  stationary:

$$\delta\Pi = \delta U - \delta W = 0 \quad \text{if} \quad u = u^*, \quad (\text{A.7})$$

---

<sup>1</sup>In Variational Calculus  $u(x)$  is called the primary variable of the functional.

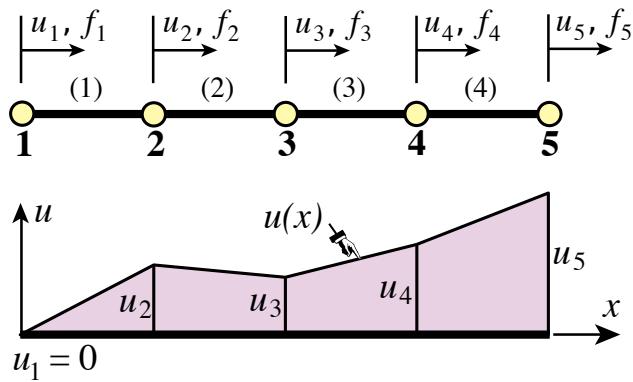


**Figure A.1:** Concept of admissible variation of the axial displacement function  $u(x)$  [23].

with respect to admissible variations  $u = u^* + \delta u$  of the exact displacement field  $u^*(x)$ . By using standard techniques of variational calculus it can be shown that if  $E A > 0$  and kinematic boundary conditions remove any rigid motion, the solution  $u^*(x)$  of Equation A.7 exists, is unique, and  $\Pi[u]$  is a minimum.

#### A.1.1.4. TPE Discretization

In order to apply the TPE functional (A.3) to the derivation of FEM equation, a replacement of the continuum mathematical model by a discrete one is needed. This new model consists of a union of bar elements.



**Figure A.2:** Fem discretization of a fixed-free bar member into four two-node elements. A piecewise-linear admissible displacement trial function  $u(x)$  is drawn under the mesh [23].

Intrinsically, functionals are scalars, and therefore, for a discretization such as the shown above, the TPE functional may be decomposed into a sum of contributions of each element,

$$\Pi = \Pi^{(1)} + \Pi^{(2)} + \dots + \Pi^{(m)}. \quad (\text{A.8})$$

where  $m$  denotes the number of elements. This decomposition applies to both the internal energy and the external work potential components,

$$\delta U = \delta U^{(1)} + \dots + \delta U^{(m)} = 0; \quad \delta W = \delta W^{(1)} + \dots + \delta W^{(m)} = 0, \quad (\text{A.9})$$

as well as to the stationary condition (Equation A.7),

$$\delta\Pi = \delta\Pi^{(1)} + \delta\Pi^{(2)} + \dots + \delta\Pi^{(m)} = 0. \quad (\text{A.10})$$

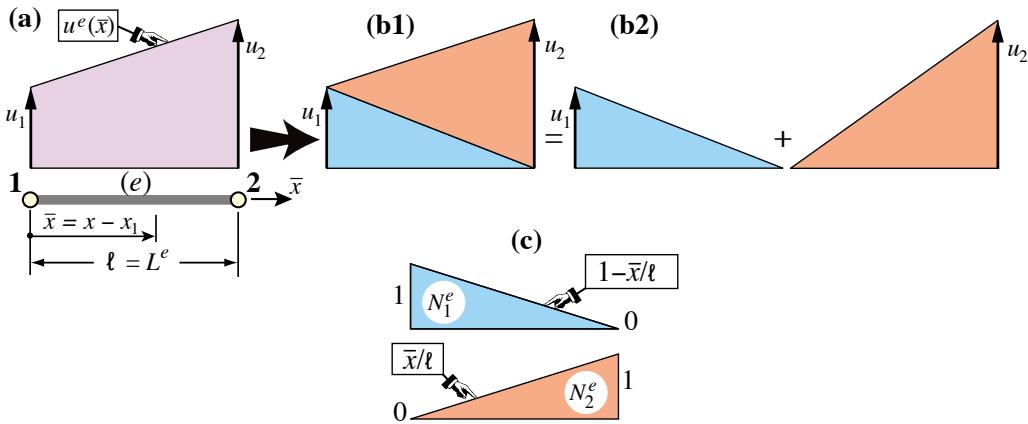
The equation above implies that for a generic element  $e$  we can write

$$\delta\Pi^e = \delta U^e - \delta W^e = 0. \quad (\text{A.11})$$

This variational equation is called a *first variation form*, and it is a special case of a more general expression called *weak form*. It is the basis for the derivation of element stiffness equations once the displacement field has been discretized over the element, as described in the following section.

#### A.1.1.5. Bar Element Discretization

Figure A.3 shows a generic bar element  $e$ . The two nodes, which are labeled 1 and 2, are called local node numbers. The element is referred to its local axis  $\bar{x}$ , and it has two degrees of freedom  $u_1^e$  and  $u_2^e$ .



**Figure A.3:** A two-node, TPE-based bar element: (a) element configuration and axial displacement variation; (b1,b2) displacement interpolation expressed in terms of linear shape functions; (c) element shape functions.

The mathematical concept of bar finite elements is to approximate axial displacement  $u(x)$  over the element. The exact displacement  $u^*$  is replaced by an approximate displacement over the finite element mesh as

$$u^* \approx u^e(x), \quad (\text{A.12})$$

where the approximate displacement  $u^e(x)$ , taken over all the elements, is called the finite element trial expansion. This trial function must belong to the class of kinematically admissible displacements defined in Section A.1.1.2. Therefore, it must be continuous over and between elements, being polynomials in  $x$  the most common choice for  $u^e$ .

#### A.1.1.6. Interpolation by Shape Functions

In the two-node bar element the only possible choice of the displacement  $u^e$  that satisfies the inter-element continuity requirement is linear [23]. It is expressed by the following interpolation equation, which is graphically expressed in Figure A.3.

$$u^e(x) = N_1^e u_1^e + N_2^e u_2^e = \begin{bmatrix} N_1^e & N_2^e \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \mathbf{N}^e \mathbf{u}^e, \quad (\text{A.13})$$

where the functions  $N_1^e$  and  $N_2^e$  that multiply the node displacements  $u_1$  and  $u_2$  are called shape functions. The shape functions the displacement  $u^e$  from the node values. For the element from Figure A.3(c), with  $\bar{x} = x - x_1$  measuring the axial distance from the left node, the shape functions are

$$N_1^e = 1 - \frac{\bar{x}}{\ell} = 1 - \zeta; \quad N_2^e = \frac{\bar{x}}{\ell} = \zeta, \quad (\text{A.14})$$

where

$$\zeta = \frac{x - x_1}{\ell} = \frac{\bar{x}}{\ell}, \quad (\text{A.15})$$

is the Greek letter  $\zeta$  represents a dimensionless coordinate, also known as a natural coordinate, that varies from 0 through 1 over the element.

#### A.1.1.7. The Strain-Displacement Matrix

The axial strain associated with the trial function  $u^e$  is computed by using the kinematic equation

$$\varepsilon = \frac{du^e}{dx} = (u^e)' = \left[ \frac{N_1^e}{dx} \quad \frac{N_2^e}{dx} \right] \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \frac{1}{\ell} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \mathbf{B} \mathbf{u}^e, \quad (\text{A.16})$$

where  $\mathbf{B}$  is the strain-displacement matrix, and it is constant over the element.

#### A.1.1.8. Basis Functions

While shape functions are associated with elements, a trial basis function, or simply basis function, is associated with a node. Suppose node  $i$  of a bar discretization connects elements  $e_1$  and  $e_2$ . The basis function  $\phi_i$  is defined as

$$\phi_i(x) \begin{cases} \phi_i^{(e1)} & \text{if } x \in \text{element } (e1) \\ \phi_i^{(e2)} & \text{if } x \in \text{element } (e2) \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.17})$$

In order to reduce the clutter and clarify the notation, the node subscript  $i$  will be added to shape functions to allow  $\phi_i = N_i^e$ .

### A.1.1.9. The Finite Element Equations

The discretization process based on the TPE functional leads to

$$\Pi^e = U^e - W^e; \quad U^e \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e; \quad W^e \stackrel{\text{def}}{=} (\mathbf{u}^e)^T \mathbf{F}^e, \quad (\text{A.18})$$

where  $\mathbf{K}^e$  and  $\mathbf{F}^e$  are called the element stiffness matrix and the element consistent nodal force vector. The three scalars  $\Pi^e$ ,  $U^e$  and  $W^e$  are only function of the node displacements  $\mathbf{u}^e$ . It is seen that  $U^e$  and  $W^e$  depend quadratically and linearly, respectively, on the node displacements  $\mathbf{u}^e$ . Taking the variation of  $\Pi^e$  with respect to the node displacements gives

$$\delta \Pi^e = (\delta \mathbf{u}^e)^T \frac{\delta \Pi^e}{\delta \mathbf{u}^e} = (\delta \mathbf{u}^e)^T [\mathbf{K}^e \mathbf{u}^e - \mathbf{F}^e] = 0. \quad (\text{A.19})$$

Owing to the arbitrary variations of  $\delta \mathbf{u}^e$ , the bracketed expression must vanish which yields to

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{F}^e. \quad (\text{A.20})$$

### A.1.1.10. Stiffness Matrix

Using the definition of internal energy  $U^e$  for the two-node bar element in A.1, and by changing the variable  $x$  to the natural coordinate  $\zeta$  the new expression leads to

$$u^e = \frac{1}{2} \int_{x_1}^{x_2} \varepsilon E A \varepsilon dx = \frac{1}{2} \int_0^1 \varepsilon E A \varepsilon \ell d\zeta. \quad (\text{A.21})$$

By using the strain-displacement matrix relation A.16, and transposing the first  $\varepsilon$  to allow the matrix multiplication, we get

$$u^e = \frac{1}{2} \int_0^1 (\mathbf{u}^e)^T \mathbf{B}^T E A \mathbf{B} \mathbf{u}^e \ell d\zeta = \frac{1}{2} \int_0^1 [u_1^e \ u_2^e] \frac{1}{\ell} \begin{bmatrix} -1 \\ 1 \end{bmatrix} E A \frac{1}{\ell} [-1 \ 1] \frac{1}{\ell} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} \ell d\zeta. \quad (\text{A.22})$$

Since  $E A$  is a scalar and knowing that the nodal displacements do not depend on position, the last expression can be rearranged as follows

$$U^e = \frac{1}{2} (\mathbf{u}^e)^T \int_0^1 E A \mathbf{B}^T B \ell d\zeta \mathbf{u}^e = \frac{1}{2} [u_1^e \ u_2^e] \int_0^1 \frac{E A}{\ell^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \ell d\zeta \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}. \quad (\text{A.23})$$

The total internal energy and work potential of the bar element is defined as

$$U^e = \frac{1}{2} (\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e; \quad W^e = (\mathbf{u}^e)^T \mathbf{F}^e. \quad (\text{A.24})$$

Now, we can extract  $\mathbf{K}^e$  from the definition of the total internal energy as

$$\mathbf{K}^e = \int_0^1 EA \mathbf{B}^T \mathbf{B} \ell d\zeta = \int_0^1 \frac{EA}{\ell^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \ell d\zeta = \frac{1}{\ell} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \int_0^1 EA d\zeta. \quad (\text{A.25})$$

Furthermore, for a homogeneous and prismatic bar of constant rigidity,  $EA$  can be moved outside the integral to give

$$\mathbf{K}^e = \frac{EA}{\ell} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \quad (\text{A.26})$$

#### A.1.1.11. Consistent Node Force Vector

The consistent node force vector, defined in A.24, comes from the element contribution to the external work potential  $W$ :

$$W^e = \int_{x_1}^{x_2} q u dx = \int_0^1 q \mathbf{N}^T \mathbf{u}^e \ell d\zeta = (\mathbf{u}^e)^T \int_0^1 q \begin{bmatrix} 1-\zeta \\ \zeta \end{bmatrix} \ell d\zeta \stackrel{\text{def}}{=} (\mathbf{u}^e)^T \mathbf{F}^e. \quad (\text{A.27})$$

Since  $\mathbf{u}^e$  is arbitrary,

$$\mathbf{F}^e = \int_{x_1}^{x_2} q \begin{bmatrix} 1-\zeta \\ \zeta \end{bmatrix} dx = \int_0^1 q \begin{bmatrix} 1-\zeta \\ \zeta \end{bmatrix} \ell d\zeta. \quad (\text{A.28})$$

## A.1.2. The Galerkin Method

The variational formulation review in the last section is based on the Minimum Total Potential Energy principle, which is a special case inside the Principle of Virtual Work used in elastic problems. This method is based on the minimization of a functional. Nevertheless, there is other kind of methods used to approximate the solution called the *Weighted Residual Methods*. In these methods, the approximation starts with an estimate of the solution and demand that its weighted average error is minimized.

The approximate solution<sup>2</sup>  $u(x)$  is constructed by using linear combinations of basis functions according to

$$u(x) = \sum_{i=1}^m N_i u_i. \quad (\text{A.29})$$

Now, a test function  $v(x)$  is defined as<sup>3</sup>

---

<sup>2</sup>The approximate displacement function  $u^e(x)$  refers  $u(x)$  to simplify notation.

<sup>3</sup>The test function is also known as weight function in the context of approximation methods.

$$v(x) = \sum_{i=1}^m N_i v_i. \quad (\text{A.30})$$

Both trial  $u(x)$  and test  $v(x)$  functions are restricted to be quadratic polynomials:

$$u(x) = a_0 + a_1 x + a_2 x^2; \quad v(x) = b_0 + b_1 x + b_2 x^2. \quad (\text{A.31})$$

In addition,  $u(x)$  and  $v(x)$  are spanned by the linear-space basis  $\{1, x, x^2\}$  of dimension 3. When we force both trial and test basis to be equal the procedure is called the *Galerkin method*.

Taking the Euler-Lagrange Equation A.5 as the residual, and introducing the latter trial function A.29 we get

$$r(x) = EA u'' + q(x). \quad (\text{A.32})$$

The above equation is called the *strong form* since the condition  $r(x) = 0$  must hold at each point over the span member  $x \in [0, L]$ . To solve this problem the *weak form* is introduced.

#### A.1.2.1. Weak Form

Weak forms occupy an intermediate position between differential equations and functionals. Sometimes insisting on  $r(x) = 0$  everywhere is too demanding. We would like to relax that condition so it is satisfied in an average sense. To accomplish that, multiply the residual by a function  $v(x)$ , integrate over the problem domain, and set the result to zero to give

$$J = \int_0^L r(x) v(x) dx = 0, \quad (\text{A.33})$$

where  $J$  is the first variation of a functional  $\Pi$ . By replacing  $r(x) = (EA u'(x) + q)'$  into the latter equation, and assuming  $EA$  constant, we get,

$$J = \int_0^L (EA u''(x) + q(x)) v(x) dx = 0. \quad (\text{A.34})$$

Weak forms never give accurate solutions because of the reduction in the requirements of smoothness and weak imposition of Neumann boundary conditions. But this comparison is valid only when you compare the weak solutions with the classical solutions. Weak forms still give relatively very accurate results with the mesh refinement, which are extremely good for engineering simulations; and you will get a solution even if there is no classical solution.

Second derivatives of  $u(x)$  appears in Equation A.34, but none of  $v(x)$ . Integration by parts restores the derivative order balance to give

$$J = \int_0^L EA u'(x) v'(x) dx - \int_0^L q(x) v(x) dx - (EA u'(x)) v(x). \quad (\text{A.35})$$

The complete process is described in the following section. Therefore, by comparing the latter equation with Equation A.4 it is seen that if we set the test function to be an admissible variation of  $u(x)$  as

$$v(x) = \delta u(x), \quad (\text{A.36})$$

both equations are the same.

#### A.1.2.2. The Finite Element Equations II

Equation A.34 reveals an asymmetry; second derivatives of  $u(x)$  appear, but not derivative  $\partial v(x)$ . This places unequal restrictions on smoothness of the trial and test function spaces [23]. Integration by parts restores derivative order balance. However, some previous work needs to be done before integrating.

This method will be generalized for a general boundary value problem. Therefore the axial load  $q$  is replaced by the body force  $f$ . First, Equation A.34 is transformed into a volume integral by assuming that area  $A$  is constant and  $A dx = dV$ . Also, by introducing the kinematic equation  $u' = \varepsilon$  and constitutive relation  $\sigma = E\varepsilon$ , Equation A.34 is rewritten as

$$\int_V v \nabla \cdot \sigma dV + \int_V v f dV = 0; \quad (\text{A.37})$$

$$\int_V v \nabla \cdot \sigma dV = - \int_V v f dV. \quad (\text{A.38})$$

Assuming a constant Young's modulus  $E$ , the elastic modulus can be extracted out of the divergence operator as

$$\int_V v E \nabla \cdot \varepsilon dV = - \int_V v f dV. \quad (\text{A.39})$$

It can be observed that the strain  $\varepsilon$  has implicitly 2 derivatives ( $\varepsilon = u'$ ). Thus, applying the derivative chain rule gives

$$\nabla \cdot (\varepsilon v) = \nabla \cdot \varepsilon v + \varepsilon \cdot \nabla v. \quad (\text{A.40})$$

Equations A.39 and A.40 have a term in common. This term can be isolated from the last equation as

$$\nabla \cdot \varepsilon v = \nabla \cdot (\varepsilon v) - \varepsilon \cdot \nabla v. \quad (\text{A.41})$$

Now Equation A.39 is arranged in the following way

$$\int_V E \nabla \cdot (\varepsilon v) dV - \int_V E \varepsilon \cdot \nabla v dV = - \int_V v f dV. \quad (\text{A.42})$$

The *Gauss' theorem*, which relates the volume and surface integral is introduced as

$$\int_V \nabla \cdot F \, dV = \int_A F \cdot n \, dA. \quad (\text{A.43})$$

Therefore, the integral of the divergence can be replaced for an evaluation at the boundary leading to

$$\int_A E\varepsilon \cdot n v \, dA - \int_V \sigma \cdot \nabla v \, dV = - \int_V v f \, dV. \quad (\text{A.44})$$

Rearranging the terms and imposing the force boundary condition  $\sigma_n = \hat{\mathbf{t}} A$ <sup>4</sup> (Seen in Section A.2), this leads to

$$\int_V \sigma \cdot \nabla v \, dV = \int_V v f + \int_A \hat{\mathbf{t}} v, \quad (\text{A.45})$$

where the above terms will now be substituted by

$$\begin{aligned} \mathbf{v} &= \mathbf{N} \mathbf{v}^e; & \nabla \mathbf{v} &= \mathbf{D} \mathbf{v} = \mathbf{D} \mathbf{N} \mathbf{v}^e = \mathbf{B} \mathbf{v}^e; \\ \mathbf{u} &= \mathbf{N} \mathbf{u}^e; & \nabla \mathbf{u} &= \mathbf{B} \mathbf{u}^e, \end{aligned} \quad (\text{A.46})$$

using these relations we get<sup>4</sup>

$$\mathbf{x}^e \int_V \mathbf{B} \boldsymbol{\sigma} \, dV = \mathbf{x}^e \int_V \mathbf{N} \mathbf{f} \, dV + \mathbf{x}^e \int_A \hat{\mathbf{t}} \mathbf{N} \, dA. \quad (\text{A.47})$$

The latter equation, the internal and external force equality is defined as

$$\int_V \mathbf{B}^T \boldsymbol{\sigma} \, dV = \int_V \mathbf{N} \mathbf{f} \, dV + \int_A \hat{\mathbf{t}} \mathbf{N} \, dA. \quad (\text{A.48})$$

In order to unveil the stiffness matrix it is necessary to define the stress as follows

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} = \mathbf{E} \mathbf{D} \mathbf{u} = \mathbf{E} \underbrace{\mathbf{D} \mathbf{N}^T}_{\mathbf{B}} \mathbf{u}^e = \mathbf{E} \mathbf{B} \mathbf{u}^e, \quad (\text{A.49})$$

Eventually, introducing the last expression into Equation A.48 yields to

$$\overbrace{\left( \int_V \mathbf{B}^T \mathbf{E} \mathbf{B} \right) \mathbf{u}^e}^{\boldsymbol{\sigma}} \, dV = \underbrace{\int_V \mathbf{N}^T \mathbf{f} \, dV + \int_A \hat{\mathbf{t}} \mathbf{N} \, dA}_{\mathbf{F}^e}. \quad (\text{A.50})$$

The finite element equation is

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{F}^e. \quad (\text{A.51})$$

---

<sup>4</sup>Bold quantities mean either vector or matrix structures.

## A.2. Plane Stress Problem

In this section the variational formulation of two-dimensional continuum finite elements. In order to introduce it, the plane stress analysis is followed.

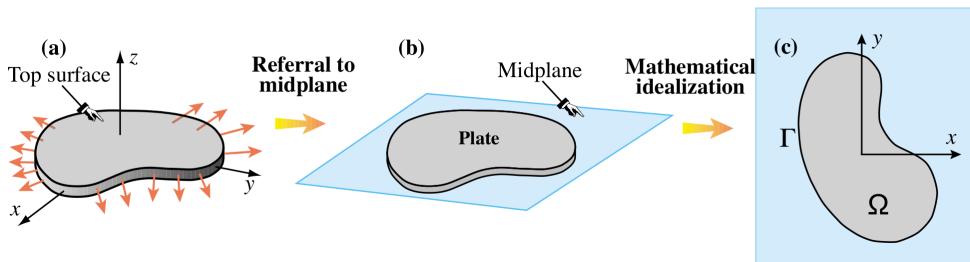
### A.2.1. Plate in Plane Stress

In structural mechanics, a flat thin sheet of material is called a plate. As continuum-based structural finite elements were invented in the aircraft industry during early 1950s [24], where plates have a major interest, this particularization will be useful for developing both the governing and finite element equations of 2D elements.

#### A.2.1.1. Assumptions

A plate loaded in its midplane<sup>5</sup> is said to be in a state of plane stress if the following assumptions are hold:

1. All loads applied to the plate act in the midplane direction, and are symmetric with respect to the midplane.
2. All support conditions are symmetric about the midplane.
3. In-plane<sup>6</sup> displacements, strains and stresses can be taken to be uniform through the thickness<sup>7</sup>.
4. The normal and shear stress components in the  $z$  direction are zero or negligible.
5. The plate is fabricated of the same material through the thickness, and thus are called homogeneous or *monocoque* plates. This assumption excludes composite and honeycomb sandwich plates which will require more complicated mathematical models [24].



**Figure A.4:** A plate structure in plane stress [24]; (a) configuration; (b) referral to its midplane; (c) 2D mathematical idealization as boundary value problem.

#### A.2.1.2. Problem Data

The mathematical model of the plane in plane stress is set up as a 2D boundary value problem (BVP). This allows to formulate the BVP over a plane domain  $\Omega$  with a boundary

<sup>5</sup>The midplane lies halfway between the two plate faces.

<sup>6</sup>In-plane directions comprise all the directions parallel to the midplane.

<sup>7</sup>In a plate, the thickness should be small, typically about 10% than the shortest in-plane dimension.

$\Gamma$ , as illustrated in Figure A.4 (b).

- **Domain geometry.** This is defined by the boundary  $\Gamma$ .
- **Thickness.** Most plates have constant thickness. Whether this thickness does vary, in which case  $h = h(x, y)$ , it should do so gradually to maintain the plate stress state.
- **Material data.** Defined by the constitutive equations. The plate material is linearly elastic and isotropic.
- **Specified Interior Forces.** These are the known forces that act in the interior  $\Omega$  of the plate. There are two types:
  - *Body or volume forces* are the forces specified per unit of volume, for example the weight.
  - *Face forces* are tangentially to the plate faces, for example the friction.
- **Specified Surface Forces.** These are the known forces that act on the boundary  $\Gamma$ .
- **Displacement Boundary Conditions.** These conditions specify how the plate is supported.

#### A.2.1.3. Problem Unknowns

The unknown fields are displacements, strain and stresses. The homogeneity assumption makes the dependence on  $z$  disappear, and all the fields become functions of  $x$  and  $y$  only.

- **Displacements.** The in-plane displacement field is defined by two components:

$$\mathbf{u}(x, y) = \begin{bmatrix} u_x(x, y) \\ u_y(x, y) \end{bmatrix}. \quad (\text{A.52})$$

The transverse displacement component  $u_z(x, y, z)$  is generally nonzero because of Poisson's ratio effects, and depends on  $z$ . However, this displacement does not appear in the governing equation [24].

- **Strains.** The in-plane strain field forms a tensor defined by three independent components.

$$\boldsymbol{\varepsilon}(x, y) = \begin{bmatrix} \varepsilon_{xx}(x, y) \\ \varepsilon_{yy}(x, y) \\ 2\varepsilon_{xy}(x, y) \end{bmatrix}. \quad (\text{A.53})$$

The factor of 2 in  $\varepsilon_{xy}$  shortens strain energy expressions, while components  $\varepsilon_{xz}$  and  $\varepsilon_{yz}$  vanish. The transverse normal strain  $\varepsilon_{zz}$  is generally nonzero also because of Poisson's ratio effects. However, this strain does not enter the governing equation since the associated  $\sigma_{zz}$  is zero.

- **Stresses.** The in-plane stress field forms a tensor defined by three independent components.

$$\boldsymbol{\sigma}(x, y) = \begin{bmatrix} \sigma_{xx}(x, y) \\ \sigma_{yy}(x, y) \\ \sigma_{xy}(x, y) \end{bmatrix}. \quad (\text{A.54})$$

### A.2.2. Governing Equations

To develop the governing equations for plane stress finite elements is done in the framework of linear elasticity. Here, the three fields are connected by the kinematic, constitutive and equilibrium equations as follows

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}; \quad (\text{A.55})$$

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{121} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} e_{xx} \\ e_{yy} \\ 2exy \end{bmatrix}; \quad (\text{A.56})$$

$$\begin{bmatrix} \partial/\partial x & 0 & \partial/\partial y \\ 0 & \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (\text{A.57})$$

These equations can be written in a matrix compact form as

$$\boldsymbol{\varepsilon} = \mathbf{D} \mathbf{u}; \quad \boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon}; \quad \mathbf{D}^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}. \quad (\text{A.58})$$

Here  $\mathbf{E} = \mathbf{E}^T$  is the matrix of plane stress elastic moduli,  $\mathbf{D}$  is the symmetric-gradient operator. Due to the material isotropic assumption, the matrix  $\mathbf{E}$  is reduced to

$$E_{11} = E_{22} = \frac{E}{(1 - \nu^2)}; \quad (\text{A.59})$$

$$E_{33} = \frac{E}{2(1 + \nu)} = G; \quad (\text{A.60})$$

$$E_{12} = \nu E_{11}; \quad (\text{A.61})$$

$$E_{13} = E_{23} = 0, \quad (\text{A.62})$$

where  $G$  is the tangent module and it is equal to the  $\mu$  Lamé constant. However, in order to generalize the constitutive equation for any type of problem, involving an isotropic material, the elasticity tensor  $\mathcal{C}$  is introduced to give

$$\boldsymbol{\sigma} = \mathcal{C} : \boldsymbol{\varepsilon}. \quad (\text{A.63})$$

Equation A.63 can also be written in terms of the Lamé coefficients as

$$\boldsymbol{\sigma} = \lambda(\operatorname{tr} \boldsymbol{\varepsilon}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}, \quad (\text{A.64})$$

where the fourth-order elasticity tensor is

$$\mathcal{C} = \lambda \mathbf{I} \otimes \mathbf{I} + \mu(\mathcal{I} + \tilde{\mathcal{I}}); \quad \mathcal{C}_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}). \quad (\text{A.65})$$

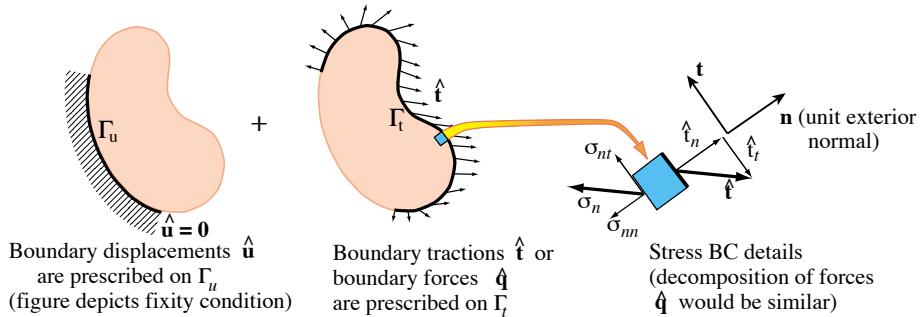
### A.2.3. Boundary Conditions

There are two types of boundary conditions (BC) prescribed on  $\Gamma$ : displacement BC or force BC. To write down those conditions it is convenient to break up  $\Gamma$  into two subsets:  $\Gamma_u$  and  $\Gamma_t$ , over which displacements and force are specified.

Displacement, or Dirichlet, boundary conditions are prescribed on  $\Gamma_u$  as

$$\mathbf{u} = \hat{\mathbf{u}}, \quad (\text{A.66})$$

where  $\hat{\mathbf{u}}$  are prescribed displacements. Often some portions of the boundary are restricted by  $\hat{\mathbf{u}} = 0$ , as the ones illustrated in figure below.



**Figure A.5:** Displacement and force (stress, traction) BC for the plane stress problem [24].

Force, or Neumann, boundary conditions are specified on  $\Gamma_t$ . They take the following form,

$$\boldsymbol{\sigma}_n = \hat{\mathbf{t}}, \quad (\text{A.67})$$

where  $\hat{\mathbf{t}}$  are prescribed surface tractions specified as a force per unit area, and  $\boldsymbol{\sigma}_n$  is the stress vector shown in Figure A.5. The components of  $\boldsymbol{\sigma}_n$  in Cartesian coordinates follow from Cauchy's stress transformation formula as

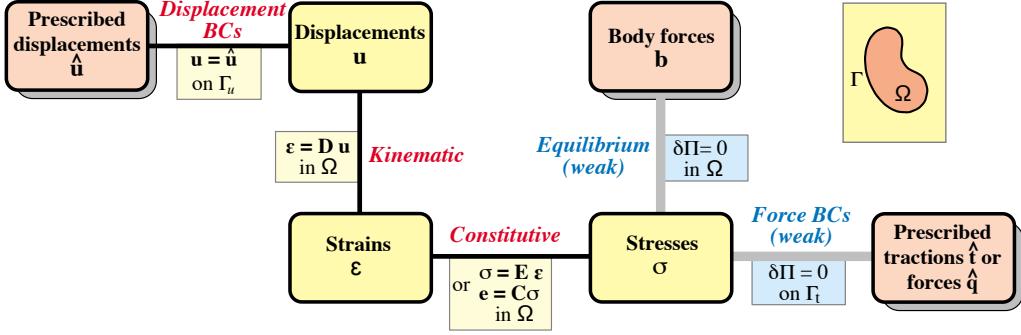
$$\sigma_{nn} = \hat{\mathbf{t}}_n; \quad (\text{A.68})$$

$$\sigma_{nt} = \hat{\mathbf{t}}_t. \quad (\text{A.69})$$

where  $n$  and  $t$  denote the normal and tangential directions respectively.

### A.2.4. Finite Element Equations

Before introducing the finite element equations, it is relevant to introduce some terminology from variational calculus introduced in Section A.1.



**Figure A.6:** The TPE-based Weak Form of the plane stress equations [24].

Figure A.6 shows the relation between the governing equations. The so-called strong and weak links are represented with black and grey lines, respectively. As it can be seen, the strong form of the equilibrium equation and the force BCs is replaced by the weak form, which enforce relations in an average or integral sense rather than point by point. These equations are given by the variational statement  $\delta \Pi = 0$ , where the TPE functional  $\Pi$  for the plane stress problem is given by  $\Pi = U - W$ .

A generic element  $e$  with  $n \geq 3$  node points has  $2n$  degrees of freedom. These DOFs are collected in the element node displacement vector in a node by node arrangement,

$$\mathbf{u}^e = [u_{x1} \ u_{y1} \ u_{x2} \ \dots \ u_{xn} \ u_{yn}]^T. \quad (\text{A.70})$$

#### A.2.4.1. Displacement Interpolation

The displacement field  $\mathbf{u}^e(x, y)$  over the element is interpolated from node displacements. We assume that the same interpolation functions are used for both displacement components  $u_x$  and  $u_y$ . Thus,

$$\mathbf{u}(x, y) = \begin{bmatrix} u_x(x, y) \\ u_y(x, y) \end{bmatrix} = \begin{bmatrix} N_1^e & 0 & N_2^e & 0 & \dots & N_n^e & 0 \\ 0 & N_1^e & 0 & N_2^e & \dots & 0 & N_n^e \end{bmatrix} \mathbf{u}^e = \mathbf{N} \mathbf{u}^e, \quad (\text{A.71})$$

where  $\mathbf{N}$  is called the shape function matrix. It has dimensions  $2 \times 2n$ . The interpolation condition on the element shape function  $N_i^e(x, y)$  states that it must take the value 1 at the  $i^{th}$  node and zero at all others. This ensures the interpolation of Equation A.71 is correct at the nodes.

Applying the kinematic relation to the finite element displacement field yields to

$$\boldsymbol{\varepsilon}(x, y) = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_2^e}{\partial x} & 0 & \dots & \frac{\partial N_n^e}{\partial x} & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & \frac{\partial N_2^e}{\partial y} & \dots & 0 & \frac{\partial N_n^e}{\partial y} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} & \dots & \frac{\partial N_n^e}{\partial y} & \frac{\partial N_n^e}{\partial x} \end{bmatrix} \mathbf{u}^e = \mathbf{B} \mathbf{u}^e, \quad (\text{A.72})$$

where  $\mathbf{B} = \mathbf{D} \mathbf{N}$  is the strain-displacement matrix. Its dimension is  $3 \times 2n$ .

#### A.2.4.2. Element Energy

To obtain the finite element stiffness equation, the variation of the TPE functional is decomposed into contributions from individual elements:

$$\delta\Pi^e = \delta U^e - \delta W^e = 0, \quad (\text{A.73})$$

where the internal energy can be expressed in terms of the strains only as

$$U^e = \frac{1}{2} \int_{\Omega} h \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} d\Omega^e = \frac{1}{2} \int_{\Omega} h \boldsymbol{\varepsilon}^T \mathbf{E} \boldsymbol{\varepsilon} d\Omega^e, \quad (\text{A.74})$$

and the external energy, which is the sum of contributions from the given interior (body) and exterior (boundary) forces, is defined as

$$W^e = \int_{\Omega} h \mathbf{u}^T \mathbf{b} d\Omega^e + \int_{\Gamma_t} h \mathbf{u}^T \hat{\mathbf{t}} d\Gamma^e. \quad (\text{A.75})$$

#### A.2.4.3. Element Stiffness

Inserting the relations  $\mathbf{u} = \mathbf{N} \mathbf{u}^e$ ,  $\boldsymbol{\varepsilon} = \mathbf{B} \mathbf{u}^e$  and  $\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon}$  into  $\Pi^e$  yields a quadratic form in the nodal displacements as

$$\Pi^e = \frac{1}{2} \mathbf{u}^{eT} \mathbf{K}^e \mathbf{u}^e - \mathbf{u}^{eT} \mathbf{F}^e. \quad (\text{A.76})$$

Here the element stiffness matrix is

$$\mathbf{K}^e = \int_{\Omega^e} h \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega^e, \quad (\text{A.77})$$

and the element nodal force vector is

$$\mathbf{F}^e = \int_{\Omega^e} h \mathbf{N}^T \mathbf{f} d\Omega^e + \int_{\Gamma^e} h \mathbf{N}^T \hat{\mathbf{t}} d\Gamma^e. \quad (\text{A.78})$$

## A.3. Three-Node Plane Stress Triangles

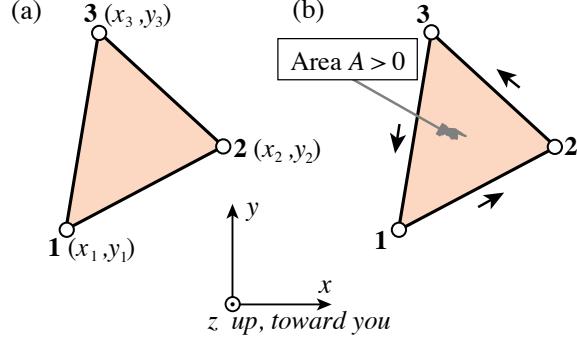
In this section the element stiffness equations of three-node triangles are derived. These elements have six displacement degrees of freedom, which are placed at the connection nodes. Only the triangles with nodes located at the corners, known as *Turner triangles*, are discussed here.

The Turner triangle has several important properties:

1. It belongs to both the isoparametric and subparametric families, which are introduced in Section A.4.
2. It allows closed form derivations for  $\mathbf{K}$  and  $\mathbf{f}$  without need for numerical integration.

3. It cannot be improved by the addition of internal DOFs.

The Turner triangle is not a good performer for structural stress analysis. It is still used in problems that do not require high accuracy, as well as in non-structural applications such as thermal and electromagnetic analysis. On the other hand, triangular meshes are easily generated over arbitrary 2D domains using techniques such as Delaunay triangulation [25].



**Figure A.7:** Three-node, linear-displacement plane stress triangular element: (a) geometry; (b) area and positive boundary transversal [25].

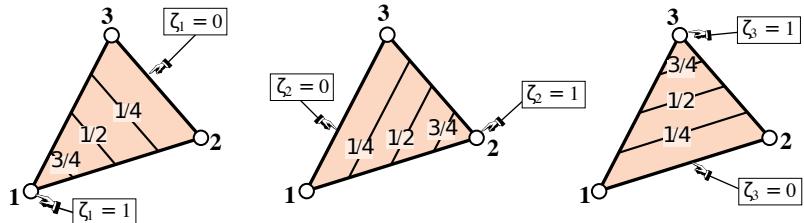
Figure A.7 shows the geometry of the Turner triangle. It has six DOFs, defined by six corner displacement components  $\{u_{xi}, u_{yi}\}$ , for  $i = 1, 2, 3$ . The triangle area can be obtained as

$$2A = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}. \quad (\text{A.79})$$

### A.3.1. Triangular Coordinates

The points of a triangle can be located in terms of a parametric coordinate system  $(\zeta_1, \zeta_2, \zeta_3)$ . These three parameters are called triangular coordinates. Due to  $\zeta_i = ct$ , each coordinate represent a set of straight lines parallel to the side opposite to the  $i^{th}$  corner, as pictured in Figure A.8. The coordinates are not independent because their sum is defined as

$$\zeta_1 + \zeta_2 + \zeta_3 = 1. \quad (\text{A.80})$$



**Figure A.8:** Triangular coordinates  $\zeta_1, \zeta_2, \zeta_3$  [25].

### A.3.2. Linear Interpolation

A function  $f(x, y)$  that varies linearly over the triangle domain in terms of Cartesian coordinates could be expressed as

$$f(x, y) = a_0 + a_1x + a_2y, \quad (\text{A.81})$$

where  $a_0$ ,  $a_1$  and  $a_2$  are coefficients to be determined from three conditions. In finite element work, these conditions are often the nodal values taken by  $f$  at the corners,  $f_1, f_2, f_3$ .

The expression in triangular coordinates lead to

$$f(\zeta_1, \zeta_2, \zeta_3) = f_1\zeta_1 + f_2\zeta_2 + f_3\zeta_3 = [f_1 \ f_2 \ f_3] \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} = [\zeta_1 \ \zeta_2 \ \zeta_3] \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}. \quad (\text{A.82})$$

The latter formula is known as linear interpolant for  $f$ .

### A.3.3. Coordinate Transformations

Quantities linked with the element geometry are best expressed in triangular coordinates. However, quantities such as displacements, strains and stresses are usually expressed in the Cartesian system. Thus, a link between both systems is needed. Both coordinates are linked by,

$$\text{Triangular} \rightarrow \text{Cartesian} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix}, \quad (\text{A.83})$$

$$\begin{aligned} \text{Cartesian} \rightarrow \text{Triangular} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} &= \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3y_1 - x_1y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1y_2 - x_2y_1 & y_1 - y_2 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \\ &= \frac{1}{2A} \begin{bmatrix} 2A_{23} & y_{23} & x_{32} \\ 2A_{31} & y_{31} & x_{13} \\ 2A_{12} & y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}, \end{aligned} \quad (\text{A.84})$$

where  $x_{jk} = x_j - x_k$ , and  $A_{jk}$  denotes the area subtended by corners  $j, k$  and the origin of the  $x-y$  system. → If the origin is at the centroid of the triangle  $A_{23} = A_{31} = A_{12} = A/3$ .

### A.3.4. Partial Derivatives

From Equations A.83 and A.84 we can get the partial derivatives as

$$\begin{aligned}\frac{\partial x}{\partial \zeta_i} &= x_i; & \frac{\partial y}{\partial \zeta_i} &= y_i; \\ 2A \frac{\partial \zeta_i}{\partial x} &= y_{jk}; & 2A \frac{\partial \zeta_i}{\partial y} &= x_{kj}.\end{aligned}\quad (\text{A.85})$$

The derivatives of a function  $f(\zeta_1, \zeta_2, \zeta_3)$  with respect to  $x$  or  $y$  follow immediately from Equation A.85 and application of the chain rule as

$$\begin{aligned}\frac{\partial f}{\partial x} &= \frac{1}{2} \left( \frac{\partial f}{\partial \zeta_1} y_{23} + \frac{\partial f}{\partial \zeta_2} y_{31} + \frac{\partial f}{\partial \zeta_3} y_{12} \right); \\ \frac{\partial f}{\partial y} &= \frac{1}{2} \left( \frac{\partial f}{\partial \zeta_1} x_{32} + \frac{\partial f}{\partial \zeta_2} x_{13} + \frac{\partial f}{\partial \zeta_3} x_{21} \right),\end{aligned}\quad (\text{A.86})$$

and in matrix form,

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \zeta_1} \\ \frac{\partial f}{\partial \zeta_2} \\ \frac{\partial f}{\partial \zeta_3} \end{bmatrix}. \quad (\text{A.87})$$

### A.3.5. Turner Triangle

The simplest triangular element for plane stress is the three-node triangle with *linear shape functions*, with degrees of freedom located at the corners. The shape functions are simply the triangular coordinates  $N_i^e = \zeta_i$ .

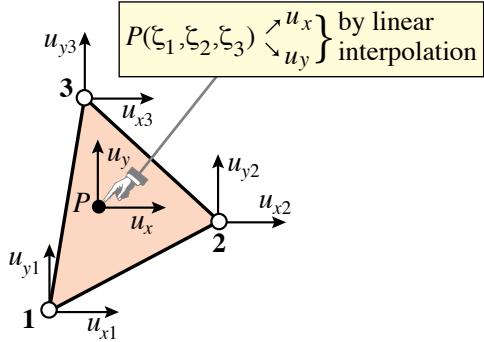
For the plane stress problem we select the linear interpolation for the displacement components  $u_x$  and  $u_y$  at an arbitrary point  $P(\zeta_1, \zeta_2, \zeta_3)$ .

$$u_x = u_{x1} \zeta_1 + u_{x2} \zeta_2 + u_{x3} \zeta_3; \quad u_y = u_{y1} \zeta_1 + u_{y2} \zeta_2 + u_{y3} \zeta_3. \quad (\text{A.88})$$

The two expressions combined lead to the following matrix,

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \zeta_1 & 0 & \zeta_2 & 0 & \zeta_3 & 0 \\ 0 & \zeta_1 & 0 & \zeta_2 & 0 & \zeta_3 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \mathbf{N} \mathbf{u}^e. \quad (\text{A.89})$$

Strains are obtained by differentiating the shape functions w.r.t.  $x$  and  $y$ .



**Figure A.9:** Displacement interpolation over triangle [25].

$$\boldsymbol{\varepsilon} = \mathbf{D} \mathbf{N} \mathbf{u}^e = \frac{1}{2A} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{21} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \mathbf{B} \mathbf{u}^e. \quad (\text{A.90})$$

Note that the strains are constant over the element. The stiffness matrix is:

$$\mathbf{K}^e = \int_{\sigma^e} h \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega. \quad (\text{A.91})$$

In triangular elements  $\mathbf{B}$  and  $\mathbf{E}$  are constant and thus,

$$\mathbf{K}^e = \mathbf{B}^T \mathbf{E} \mathbf{B} \int_{\Omega^e} h d\Omega. \quad (\text{A.92})$$

If  $h$  is uniform over the element the remaining integral is simply  $h A$ , and we obtain:

$$\mathbf{K}^e = A h \mathbf{B}^T \mathbf{E} \mathbf{B}. \quad (\text{A.93})$$

For simplicity we consider here only internal body forces,

$$\mathbf{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \quad (\text{A.94})$$

which is specified per unit of volume. The consistent nodal force vector  $\mathbf{F}^e$  is given by the following expression:

$$\mathbf{F}^e = \int_{\Omega^e} h \mathbf{N}^T \mathbf{b} d\Omega = \int_{\Omega^e} h \begin{bmatrix} \zeta_1 & 0 \\ 0 & \zeta_1 \\ \zeta_2 & 0 \\ 0 & \zeta_2 \\ \zeta_3 & 0 \\ 0 & \zeta_3 \end{bmatrix} \mathbf{f} d\Omega. \quad (\text{A.95})$$

The simplest case is when the body force components and thickness  $h$  are constant, and then the integrals

$$\int_{\Omega^e} \zeta_1 d\Omega = \int_{\Omega^e} \zeta_2 d\Omega = \int_{\Omega^e} \zeta_3 d\Omega = \frac{1}{3} A, \quad (\text{A.96})$$

which replaced yields

$$\mathbf{F}^e = \frac{A h}{3} [f_x \ f_y \ f_x \ f_y \ f_x \ f_y]^T. \quad (\text{A.97})$$

### A.3.6. Isoparametric Triangles

The isoparametric representation will be addressed in the next section. However, here is introduced for triangular elements. The key idea of isoparametric representation is to represent both the element geometry and the problem unknowns by using the shape functions. Therefore, this property will replace the *triangular coordinates* by a more general one called *natural coordinates*.

Two different types of triangles: linear and quadratic are expressed below in terms of natural coordinates.

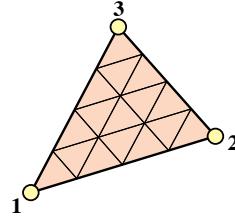
#### A.3.6.1. Linear Triangle

The 3-noded linear triangle (Figure A.10) can be represented as an isoparametric element,

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ u_{x1} & u_{x2} & u_{x3} \\ u_{y1} & u_{y2} & u_{y3} \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ N_3^e \end{bmatrix}. \quad (\text{A.98})$$

The shape functions are simply the triangular coordinates:

$$N_1^e = \zeta_1; \quad N_2^e = \zeta_2; \quad N_3^e = \zeta_3. \quad (\text{A.99})$$

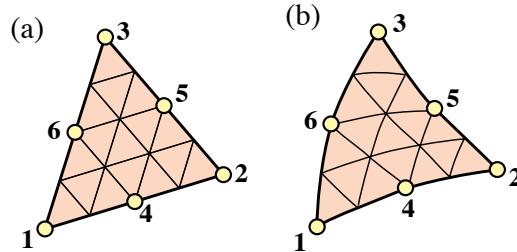


**Figure A.10:** The 3-node linear triangle [26].

#### A.3.6.2. Quadratic Triangle

The 6-node triangle shown in Figure A.11 is the next complete-polynomial member of the isoparametric triangle family. The isoparametric definition is

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ u_{x1} & u_{x2} & u_{x3} & u_{x4} & u_{x5} & u_{x6} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} & u_{y5} & u_{y6} \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ N_3^e \\ N_4^e \\ N_5^e \\ N_6^e \end{bmatrix}. \quad (\text{A.100})$$



**Figure A.11:** The 6-node quadratic triangle: (a) the superparametric version; (b) the isoparametric version [26].

The shape functions are:

$$N_1^e = \zeta_1(2\zeta_1 - 1); \quad N_2^e = \zeta_2(2\zeta_2 - 1); \quad N_3^e = \zeta_3(2\zeta_3 - 1); \quad (\text{A.101})$$

$$N_4^e = 4\zeta_1\zeta_2; \quad N_5^e = 4\zeta_2\zeta_3; \quad N_6^e = 4\zeta_3\zeta_1. \quad (\text{A.102})$$

## A.4. Isoparametric Quadrilaterals

### A.4.1. Isoparametric Representation

The stiffness equations of the linear triangle can be extended to quadrilateral elements, but encountering some technical difficulties:

- Shape functions for higher order elements with curved boundaries becomes increasingly complicated.
- Integral that appear in the expressions of the element stiffness matrix  $\mathbf{K}^e$  and consistent nodal force vector  $\mathbf{f}^e$  can no longer be evaluated in simple closed form.

These two obstacles can be overcome through the concept of *isoparametric elements* and *numerical quadrature*, respectively. The two key equations define the triangle geometry and the primary variable (in this case displacement), and they are expressed as

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix}, \quad (\text{A.103})$$

$$u_x = u_{x1} N_1^e + u_{x2} N_2^e + u_{x3} N_3^e = u_{x1} \zeta_1 + u_{x2} \zeta_2 + u_{x3} \zeta_3, \quad (\text{A.104})$$

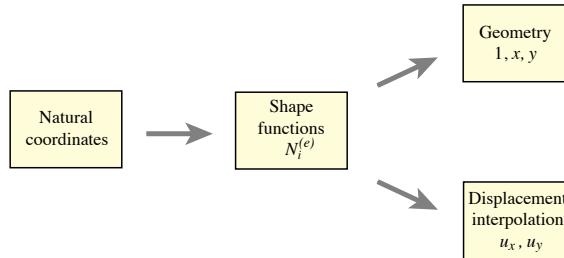
$$u_y = u_{y1} N_1^e + u_{y2} N_2^e + u_{y3} N_3^e = u_{y1} \zeta_1 + u_{y2} \zeta_2 + u_{y3} \zeta_3. \quad (\text{A.105})$$

Triangular coordinates define element geometry (Equation A.103) while displacements (Equation A.104) are defined by shape functions, which are in turn expressed in terms of natural coordinates (triangular coordinates for triangles, quadrilateral coordinates for quadrilaterals). Equations A.103 and A.104 do not look alike, but they have an inherent similarity if we rewrite the second and ajoined to Eq. A.103.

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ u_{x1} & u_{x2} & u_{x3} \\ u_{y1} & u_{y2} & u_{y3} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ u_{x1} & u_{x2} & u_{x3} \\ u_{y1} & u_{y2} & u_{y3} \end{bmatrix} = \begin{bmatrix} N_1^e \\ N_2^e \\ N_3^e \end{bmatrix}. \quad (\text{A.106})$$

This form emphasizes that geometry and displacements are given by the same parametric representation.

The key idea is to use the shape functions to represent both the element geometry and the problem unknowns (displacements). Hence the isoparametric element uses the shape functions connect the geometry with the displacements. This generalization is shown in Figure A.12



**Figure A.12:** Isoparametric representation of arbitrary 2D elements: triangles or quadrilaterals. For 3D elements, expand the geometry list to  $\{1, x, y, z\}$  and the displacements to  $\{u_x, u_y, u_z\}$  [26].

### A.4.2. Quadrilateral Coordinates

Two set of relations, one for the element geometry, and other for the element displacements are required.

*Geometric relations:*

$$1 = \sum_{i=1}^n N_i^e; \quad x = \sum_{i=1}^n x_i N_i^e; \quad y = \sum_{i=1}^n y_i N_i^e, \quad (\text{A.107})$$

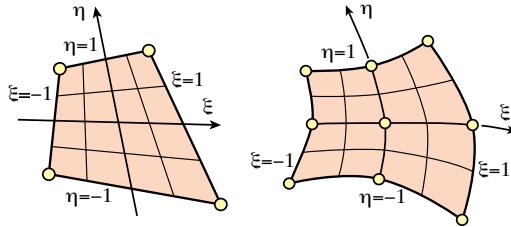
*Displacement interpolation:*

$$u_x = \sum_{i=1}^n u_{xi} N_i^e \quad u_y = \sum_{i=1}^n u_{yi} N_i^e. \quad (\text{A.108})$$

These two sets of equations can be combined in matrix form as

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ u_{x1} & u_{x2} & \dots & u_{xn} \\ u_{y1} & u_{y2} & \dots & u_{yn} \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ \vdots \\ N_n^e \end{bmatrix}. \quad (\text{A.109})$$

The natural coordinates for a triangular element are the triangular coordinates  $(\zeta_1, \zeta_2, \zeta_3)$ . The natural coordinates for a quadrilateral element are  $\xi$  and  $\eta$ , which are illustrated in Figure A.13 for both straight sided and curved side quadrilaterals.



**Figure A.13:** Quadrilateral coordinates [26].

These coordinates vary from  $-1$  to  $+1$ . In some FEM derivations it is convenient to visualize the quadrilateral coordinates plotted as Cartesian coordinates in the  $\{\xi, \eta\}$  plane, called *reference plane*. All quadrilateral elements in the reference plane become a square of side 2, called the *reference element*.

The transformation between  $\{\xi, \eta\}$  and  $\{x, y\}$  is called the *isoparametric mapping*.

### A.4.3. Bilinear Quadrilateral

The 4-node quadrilateral shown in Figure A.13 (left) is the simplest member of the quadrilateral family. It is defined by:

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ u_{x1} & u_{x2} & u_{x3} & u_{x4} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ N_3^e \\ N_4^e \end{bmatrix}. \quad (\text{A.110})$$

The shape functions are:

$$N_1^e = \frac{1}{4}(1 - \xi)(1 - \eta); \quad N_2^e = \frac{1}{4}(1 + \xi)(1 - \eta), \quad (\text{A.111})$$

$$N_3^e = \frac{1}{4}(1 + \xi)(1 + \eta); \quad N_4^e = \frac{1}{4}(1 - \xi)(1 + \eta). \quad (\text{A.112})$$

These functions vary linearly on quadrilateral coordinate lines ( $\xi$  and  $\eta$  are constant), but are not linear polynomials as in the case of the three-node triangle.

#### A.4.4. Partial Derivative Computation

To construct *quadrilateral elements* for the plane stress problem we need to follow the next steps:

- Construction of shape functions.
- Computations of shape function derivatives to compute the strain-displacement matrix  $e = \mathbf{B} \mathbf{u}^e$ .
- Numerical integration over the element using the Gauss quadrature rules.

Assuming the shape functions have been constructed, the second and third items are combined in an algorithm suitable for programming any isoparametric quadrilateral.

#### A.4.5. Partial Derivatives

Partial derivatives of shape functions w.r.t. Cartesian coordinates are required, but shape functions are not directly functions of  $x$  and  $y$  (for a isoparametric quadrilateral), but of the natural coordinates  $\xi$  and  $\eta$ .

The *Jacobian*  $\mathbf{J}$  will be needed. It connects the differentials of  $\{x, y\}$  with  $\{\xi, \eta\}$  and vice-versa.

$$BJ = \frac{\partial(x, y)}{\partial(\xi, \eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}, \quad (\text{A.113})$$

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}; \quad \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} dx \\ dy \end{bmatrix}. \quad (\text{A.114})$$

The scalar symbol  $J$  is reserved for the determinant of  $\mathbf{J}$ . In 1D  $\mathbf{J}$  and  $J$  coalesce. The derivatives of the shape functions for a quadrilateral element are, using the chain rule, the following:

$$\frac{\partial N_i^e}{\partial x} = \frac{\partial N_i^e}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial \eta}{\partial x}; \quad \frac{\partial N_i^e}{\partial y} = \frac{\partial N_i^e}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial \eta}{\partial y}. \quad (\text{A.115})$$

In matrix form,

$$\begin{bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix}. \quad (\text{A.116})$$

To compute  $\mathbf{J}$  we use the geometric relations:

$$x = \sum_{i=1}^n x_i N_i^e \quad y = \sum_{i=1}^n y_i N_i^e, \quad (\text{A.117})$$

and we differentiate w.r.t. the quadrilateral coordinates:

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^n x_i \frac{\partial N_i^e}{\partial \xi}; \quad \frac{\partial y}{\partial \xi} = \sum_{i=1}^n y_i \frac{\partial N_i^e}{\partial \xi}, \quad (\text{A.118})$$

$$\frac{\partial x}{\partial \eta} = \sum_{i=1}^n x_i \frac{\partial N_i^e}{\partial \eta}; \quad \frac{\partial y}{\partial \eta} = \sum_{i=1}^n y_i \frac{\partial N_i^e}{\partial \eta}, \quad (\text{A.119})$$

because the  $x_i$  and  $y_i$  do not depend on  $\xi$  and  $\eta$ ,

$$\begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1^e}{\partial \xi} & \frac{\partial N_2^e}{\partial \xi} & \cdots & \frac{\partial N_n^e}{\partial \xi} \\ \frac{\partial N_1^e}{\partial \eta} & \frac{\partial N_2^e}{\partial \eta} & \cdots & \frac{\partial N_n^e}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}, \quad (\text{A.120})$$

$$\mathbf{J} = \mathbf{P} \mathbf{X}. \quad (\text{A.121})$$

A quadrilateral element is said to have a *constant metric* (CM) if  $J = \det(\mathbf{J})$  is constant over its domain, such as rectangles and parallelograms. Otherwise it has *variable metric* (VM). In the latter case  $J$  is a rational function of the natural coordinates.

The nonzero entries of  $\mathbf{B}$  are calculated by following the next steps:

1. Compute the Jacobian  $\mathbf{J}$  using Equation A.120.
2. Invert the Jacobian  $\mathbf{J}^{-1}$ .
3. Use the chain rule Equation A.116.

## A.4.6. Numerical Integration

Numerical integration is an essential tool for practical evaluation of integrals over *isoparametric* element domains of variable metric (VM). Since 1960s the standard practice has been to use *Gauss integration* because such rules use a minimal number of sample points to achieve a desired level of accuracy, which is important for efficient calculations since  $\mathbf{B}^T \mathbf{E} \mathbf{B}$  is evaluated at each Gauss sample point.

Points	Rule
1	$\int_{-1}^1 F(\xi) d\xi \approx 2F(0)$
2	$\int_{-1}^1 F(\xi) d\xi \approx F(-1/\sqrt{3}) + F(1/\sqrt{3})$
3	$\int_{-1}^1 F(\xi) d\xi \approx \frac{5}{9}F(-\sqrt{3/5}) + \frac{8}{9}F(0) + \frac{5}{9}F(\sqrt{3/5})$

**Table A.1:** One-dimensional Gauss rules with 1 through 3 sample points.

Gauss product rules for quadrilateral are divided depending on the dimension.

### A.4.6.1. 1D rules

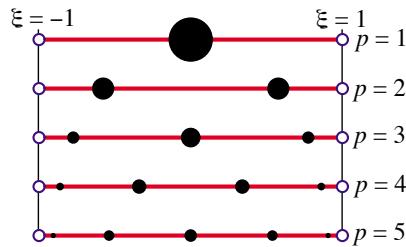
The classical Gauss integration rules are defined by

$$\int_{-1}^1 F(\xi) d\xi \sim \sum_{i=1}^p \omega_i F(\xi_i), \quad (\text{A.122})$$

where  $p \geq 1$  is the number of Gauss integration points (also known as sample points),  $\omega_i$  are the integration weights, and  $\xi_i$  are the sample-point abscissas  $\in [-1, 1]$ . The use of the **canonical** interval  $[-1, 1]$  is no restriction because an integral over another range e.g. from  $a$  to  $b$  can be transformed to the canonical interval via a simple linear transformation of the independent variable as

$$\int_a^b F(x) dx = \int_{-1}^1 F(\xi) J d\xi = \int_{-1}^1 F(\xi) \frac{1}{2} \ell d\xi. \quad (\text{A.123})$$

In general a 1D Gauss rule with  $p$  points integrates exactly polynomials of order up to  $2p - 1$ , called the *degree* of the formula.



**Figure A.14:** First five 1D Gauss rules pictured over the segment  $\xi \in [-1, +1]$  [27].

#### A.4.6.2. 2D rules

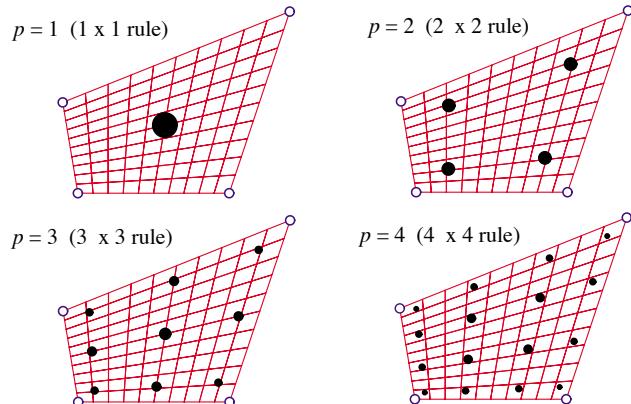
The simplest bidimensional (2D) Gauss integration formulas are called *product rules*. They are obtained applying the 1D rules along each natural coordinate in turn. To use them we must first reduce the integrand to the canonical form:

$$\int_{-1}^1 \int_{-1}^1 F(\xi, \eta) d\xi d\eta = \int_{-1}^1 d\eta \int_{-1}^1 F(\xi, \eta) d\xi, \quad (\text{A.124})$$

then we can process numerically each integral in turn:

$$\int_{-1}^1 \int_{-1}^1 F(\xi, \eta) d\xi d\eta \approx \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \omega_i \omega_j F(\xi_i, \eta_j) = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \omega_{ij} F(\xi_i, \eta_j). \quad (\text{A.125})$$

Here,  $p_1$  and  $p_2$  denote the number of Gauss points in the  $\xi$  and  $\eta$  directions. Usually the same number  $p = p_1 = p_2$  is chosen if the shape functions are taken to be the same in the  $\xi$  and  $\eta$  directions. This is the case of all quadrilaterals of Fig A.15.



**Figure A.15:** First four 2D Gauss rules  $P = 1, 2, 3, 4$  pictured over a straight-sided quadrilateral region.

#### A.4.6.3. Stiffness matrix

The stiffness matrix of a general plane stress element is given by

$$\mathbf{K}^e = \int_{\Omega^e} h \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega^e. \quad (\text{A.126})$$

A constant thickness  $h$  may be pulled out of the integral. A variable  $h$  may be interpolated from node values using the element shape functions. The elasticity matrix  $\mathbf{E}$  is usually constant. To integrate Equation A.126 numerically by a 2D product Gauss rule, it must be first put in the canonical form:

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 \mathbf{F}(\xi, \eta) d\xi d\eta. \quad (\text{A.127})$$

Everything fits the previous equation since  $\mathbf{B}$  is a function of the natural coordinates while  $h$  and  $\mathbf{E}$  may be interpolated if necessary. The exception is the area differential  $d\Omega^e \rightarrow$  we need to express as  $\rightarrow d\xi d\eta$

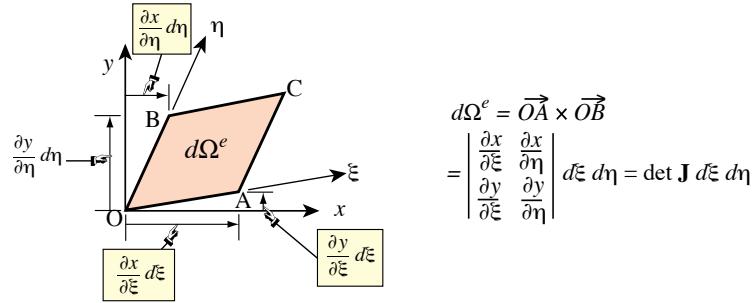
$$d\Omega^e = dx dy = J d\xi d\eta, \quad (\text{A.128})$$

the canonicalized integrand is

$$\mathbf{F}(\xi, \eta) = h \mathbf{B}^T \mathbf{E} \mathbf{B} J. \quad (\text{A.129})$$

If the same number of Gauss points  $p$  are used in the  $\xi$  and  $\eta$  directions we have:

$$\mathbf{K}^e = \sum_{i=1}^p \sum_{j=1}^p \omega_{ij} h(\xi_i, \eta_j) \mathbf{B}(\xi_i, \eta_j)^T \mathbf{E} \mathbf{B}(\xi_i, \eta_j) \det \mathbf{J}(\xi_i, \eta_j). \quad (\text{A.130})$$



**Figure A.16:** Geometric interpretation of the Jacobian determinant role [27].