UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Analysis of Slender Aircraft Structures using the Finite Element Method

Filipe José Robalo Fraqueiro

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(Ciclo de estudos integrado)

Orientador: Prof. Pedro Vieira Gamboa

Covilhã, junho de 2018

To my dear Parents.

Acknowledgement

I want to thank my parents for supporting me during these last years and for never having doubts about my capabilities. I also want to thank Professor Pedro Vieira Gamboa for having supervising me during this work and for the knowledge that he passed to me. And thank to my other professors during this course. Finally I want to thank my friends and a special thanks to Pedro Albuquerque that helped me to improve some important soft skills.

Resumo

Actualmente há uma grande exigência para que as análises estruturais sejam o mais realistas possiveis e altamente fiáveis. Na indústria aeronáutica isto é traduzido numa redução de peso das aeronaves mantendo os factores de segurança desejados. Porém este tipo de análises são um processo demorado e trabalhoso e por vezes é necessário apresentar uma solução mais rápida, embora menos fiável, que possa ser utilizada como ponto de partida no projecto preliminar.

Existem diversos métodos para resolver os diferentes problemas estruturais. Quando os casos são simples é possivel chegar a uma solução analítica, porém estes casos são muito reduzidos e envolvem a resolução de equações diferenciais. Assim é necessário recorrer a métodos numéricos de interpolação ou de aproximação. Um destes métodos é o método dos elementos finitos. Este método consiste em dividir o domínio do problema em pequenas fracções, chamadas de elementos, e aplicar aproximações em cada uma dessas fracções tendo em consideração os resultados das fracções envolventes. No caso desta dissertação este método é aplicado do ponto de vista unidimensional (1D), ou seja, as estruturas são do tipo: viga, barra de tracçao e barra de torção e os elementos são representados por segmentos de recta. Os pontos inicias e finais destes segmentos são chamados de nós e permitem a continuidade da estrutura estando conectados aos elementos adjacentes.

Duas grandes tarefas foram executadas neste trabalho. Inicialmente o método matemático dos elementos finitos foi ajustado para uma análise estrutural, mais concretamente em estruturas que possam ser consideradas em uma dimensão, e para determinar a deformação da estrutura sob determinados carregamentos e condições de fronteira. É ainda possivel calcular os modos de vibração livre da estrutura. Depois este método foi aplicado num algoritmo de forma a poder ser programado e aplicado a diversos casos.

De modo a verificar a fiabilidade do código desenvolvido foram estudados alguns casos com diferentes dados iniciais. Estes dados iniciais referem-se: ao tamanho e perfil da estrutura, material, cargas aplicadas e condições de fronteira, sendo os resultados comparados com programas já existentes no mercado, verificando assim se são plausíveis.

Palavras-chave

Análise Estrutural, Método dos Elementos Finitos, Viga, Condições de Fronteira, Equações Diferenciais

Abstract

At present there is a great demand for structural analysis to be as realistic as possible and highly reliable. In the aeronautical industry this is translated into a reduction of the aircrafts weight and maintaining the desired safety factors. However, this type of analysis is a time-consuming and labor-intensive process and sometimes it is necessary to have a faster, but less reliable, solution that can be used as a starting point in the preliminary design.

There are several methods for solving the different structural problems. When the cases are simple it is possible to have an analytical solution, however these cases are reduced and involves the resolution of differential equations. Thus it is necessary to use numerical methods of interpolation or approximation. One of these methods is the finite element method. This method consists in dividing the problem domain into small fractions, called elements, and applying approximations in each of these fractions taking into account the results of the surrounding fractions. In the case of this dissertation this method is applied from a one-dimensional (1D) point of view, ie, the structures are of the type: beam, bar and rod and the elements are represented by straight line segments. The start and end points of these segments are called nodes and allow the continuity of the structure and they are connected to the adjacent elements.

Two major tasks were performed in this work. Initially the mathematical method of the finite elements was adjusted for a structural analysis, more concretely in structures that can be considered in one dimension, and to determine the deformation of the structure under certain loads and boundary conditions. It is still possible to calculate the free vibration modes of the structure. Then this method was applied to an algorithm so that it could be programmed and applied to several cases.

In order to verify the reliability of the developed code some cases with different initial data were studied. These initial data refers to: the size and cross-section of the structure, material, loads and boundary conditions. The results are compared with programs that already exist in the market, thus verifying if they are plausible.

Keywords

Structural Analysis, Finite Element Method, Beam, Boundary Conditions, Differential Equations

Contents

List of Figures

List of Tables

List of Acronyms

BC	Boundary Conditions
CAD	Computer Aided Design
DE	Differential Equation
DOF	Degrees of Freedom
EO	Equation of Motion
FDM	Finite Differences Method
FEA	Finite Element Analysis
FEM	Finite Element Method
ID	Identification Number
IE	Integral Equation
PVW	Principle of Virtual Work
1D	One Dimension
3D	Three Dimension

List of Symbols

A	Cross section area
C	Constitutive matrix
cu	Compressive strength
D	Differentiation matrix
d	Approximation displacements vector
E	Young's modulus
e	Element
EA	Axial stiffness
EI	Bending stiffness
F	Loads matrix
G	Shear modulus
GJ	Torsional stiffness
I_p	Inertial polar moment
I_ψ	Warping moment
i	x-axis unit vector
J	Torsional Constant
j	y-axis unit vector
k	z-axis unit vector
K	Rigidity matrix
l	Length
m	Mass
M_x	Concentrated torsion momentum
M_y	Concentrated bending momentum
M_z	Concentrated bending momentum
N	Number of elements
N	Approximation Local coordinates vector
P_x	Concentrated normal force
P_y	Concentrated shear force
P_z	Concentrated shear force
Q_x	Distributed normal force
Q_y	Distributed shear force
Q_z	Distributed shear force
rot_e	Element rotation matrix
s	Displacements vector
t	Time
tu	Tensile strength
u	Linear displacement along x direction
v	Linear displacement along y direction
w	Linear displacement along z direction
W_{ext}	External Work
$W_{ext_{inertia}}$	External inertia Work
$W_{ext_{inertia}}$	External loads Work

W_{int}	Internal Work
W_x	Distributed torsion momentum
W_y	Distributed bending momentum
W_z	Distributed bending momentum
$\overline{X}$	Space coordinates vector
x	x-axis coordinate
y	y-axis coordinate
z	z-axis coordinate

List of Greek Symbols

γ	Shear strain
ε	Normal strain
$\overline{\varepsilon}$	Strain matrix
$\widehat{\varepsilon}$	Strain vector
θ_x	Angular displacement along x direction
θ_y	Angular displacement along y direction
θ_z	Angular displacement along z direction
ν	Eigenvector
ν	Poisson's ratio
ξ	Local variable
ρ	Density
σ	Normal Stress
$\overline{\sigma}$	Stress matrix
$\widehat{\sigma}$	Stress vector
τ	Shear stress
ϕ_{h_e}	Horizontal rotation angle
ϕ_{v_e}	Vertical rotation angle
ϕ_{x_e}	x-axis rotation angle
ψ	Warping function
ω	Eigenvalue

Chapter 1

Introduction

1.1 Motivation

One of the most important objectives in aircraft design and other engineering activities is to achieve the most reliable product with the lowest possible costs. So, it is very important to perform accurate calculations using the minimum computational resources possible. In structural analysis, this accurate calculation can result in a weight reduction and in aerospace engineering this means that the aircraft can be more efficient and have lower manufacturing and operational costs. The use of less resources is also important to preserve the planet environment.

In the present day there are some computer software packages that perform finite element analysis (FEA) such as ANSYS [ANS18], Dassault Systèmes ABAQUS [Das18] and MSPATRAN [MSC18]. These software packages have years, or even decades, of development and perform multiple types of analyses. However, to legally use those commercially available software packages a paid license is required. Another disadvantage is that the user must have some knowledge and experience to use these software because they have a large number of available models and tools.

Considering this disadvantages it would be very useful a simple and user friendly tool, that allows a simple and fast structural analysis to calculate structures displacements. This tool aims to be used in structures that could be assumed with just one dimension (1D) and allow all the six space degrees of freedom (DOF).

Analysing propellers and wing spars to check if them resist and deform to flight loads are some applications that motivate the development of the present work. This applications should also have analysing-time benefits because this tool should perform quick analysis. Other motivation for the development of a structural analysis code is that it could be incorporated in an optimization problem. For example, in propeller structural analysis there is the possibility of coupling this code to an aerodynamics analysis code where the loads that result from this last code can be applied in the structural one and the result is a new shaped propeller that can be again subjected to the aerodynamics code. This are commonly known as aerostructural analysis. As a low-fidelity method it prove to be superior over their high-fidelity counterparts when a high number of model evaluations are required as for instance when performing a design optimization since high-fidelity analyses are often very time-consuming, like [JS14] explained.

There are several applications of the finite element method (FEM) using 1D elements in civil, naval and aeronautical structures. In naval structures [EC16] uses 1D beam models claiming that those are much simpler compared to three dimensions (3D) solid FEM. Also, [EC13] uses 1D models, for the same reason presented before, when the objective is to analyse slender bodies, such as columns, rotor-blades, aircraft wings, towers and bridges. The both use 1D

elements formulated with Euler–Bernoulli and Timoshenko beam models to determine structures deformations and free vibration modes. Although one limitation with the Euler–Bernoulli beam models is that they ignore the transverse shear stresses or, in a best option, it is assumed as constant over the beam cross-section. The 1D element theories do not capture cross-section warping.

1.2 Objectives

The main objective of this thesis is to develop a simple and user-friendly tool to perform FEA in one dimension so that a preliminary study of the stiffness characteristics of a beam-like slender structure can be performed in an efficient and accurate way. This FEA is applied to structural engineering and in this case, it calculates the six displacements in a cartesian reference frame: three axial displacements and three angular displacements. It also aims to allow the computation of the structure free vibration modes. The meaning of 1D analysis is that the code should be able to recognise structures of beam type, in order words, one of the structure dimensions is relatively larger than the other two. This tool consists in a Fortran [Wik18] code that reads the required inputs from a text file. It also needs to be tested in order to satisfy the mechanical requirements of the propellers analysed in [Mor16] and guarantee its reliability.

Furthermore, a validation of the tool should be performed, through the comparison with a mature software that have years of development like Ansys.

1.3 Thesis Outline

To accomplish the objectives, the first task is to study the mathematical model of the finite element method for the problem described. After studying this model, it is possible to check the required analysis variables that produce the desired results. Then it is fundamental to apply a coherent algorithm that translates the results of the mathematical model. After these two steps, some simple cases are selected to verify if the results are plausible and compare them with other available software solutions. Finally, a case study can be performed to complete the code validation.

Chapter 2 features a brief history of finite element analysis background. This chapter allows the understanding of the power of this method and provides a brief theoretical explanation of how it is applied in this context. It also explains the mathematical model that supports the method. The results of this model are the code algorithm and inputs. It also provides a bridge between the structures physical properties and its deformations when subjected to certain loads.

Chapter 3 describes the computational code and the details about the cross-section properties, that are important to study beam type structures.

Chapter 4 presents the work done to validate the code. To validate the code some structures are selected and then are applied to the developed code, those structures are also analysed in ANSYS software. Then both results are compared, calculating the deviation between both,

after that the differences and similarities are discussed.

Chapter 5, similarly to chapter 4, describes a case study to validate the code. In this case study the code inputs are five blades, designed in [Mor16], with different geometries, materials and loads. The results from those analysis are compared with the ones obtained in CATIA V5 software by [Mor16] and Ansys.

Chapter 6 presents the general discussion and conclusions of the work developed during the current dissertation and some possible future works to achieve better results.

Chapter 2

The Finite Element Method

This chapter show the maths that formulates the finite element method. As explained in the previous chapter, this method can be used in various fields of physics, maths and engineering. So it is necessary adapt the method to this thesis case: displacements of 1D structures. In order to do that it is first necessary to establish the relations between this displacements and the extensions and tensions that the structure can experience. Then, three different 1D structures are considered: rod, bar and beam, the last one is also divided in two. This is required to study the different displacements, once each one has different behaviours for the different loads directions. The 1D elements for the different types are formed by two tip nodes and the displacements variations along the elements is linear.

2.1 Historical Overview

Before developing a practical tool, it is fundamental to understand the theoretical principle of the finite element method and its origins. A brief historical description of the development of this analysis methodology is present in this chapter.

In physics and engineering there are a lot of problems that are mathematically expressed using differential equations (DE) and integral equations (IE). Problems such as fluid flows, heat transfer, mass transfer, vibrations of structures, electromagnetic interactions and stress and strain of structures [Bat14] are represented by complex equations and require different methods to be solved. Some simple equations have an exact solution and are solvable by explicit formulas. However, most DE, and the ones presented in this dissertation, require numerical methods to be solved. Figure 2.1 resumes some methods to solve analytic and numeric DE.

One of the first methodologies to solve DE was the finite differences method (FDM). This represents a group of numerical methods to solve the DE that uses approximations in the derivatives of the differential equations. Other techniques like various weighted residual procedures or approximate techniques for determining the stationary of a properly defined functional were also developed by mathematicians, as Figure 2.2 shows. All these discretization methods use approximations to represent a continuum solution but, they were not intuitively enough for some problems. So, engineers found a more intuitive way to solve them by creating an analogy between real discrete elements and finite portions of a continuum domain [OCZ00]. This was called finite element method.

A vast number of engineers, physicists and mathematicians had contributed for the development and evolution of this methodology. According to [Bat14], the original contributions appeared in the papers by J. H. Argyris and S. Kelsey; M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp, and R. W. Clough in 1950s. The first and perhaps the simplest definition of the

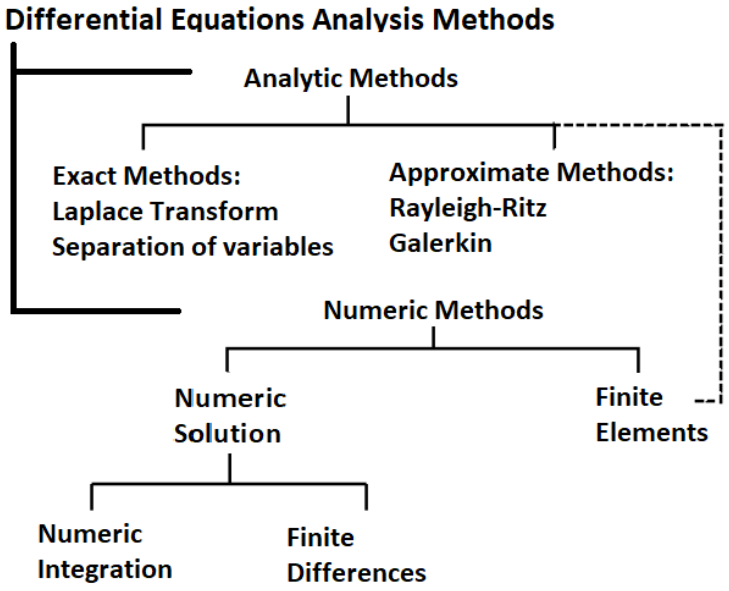

Figure 2.1: Differential Equations Analysis Methods.

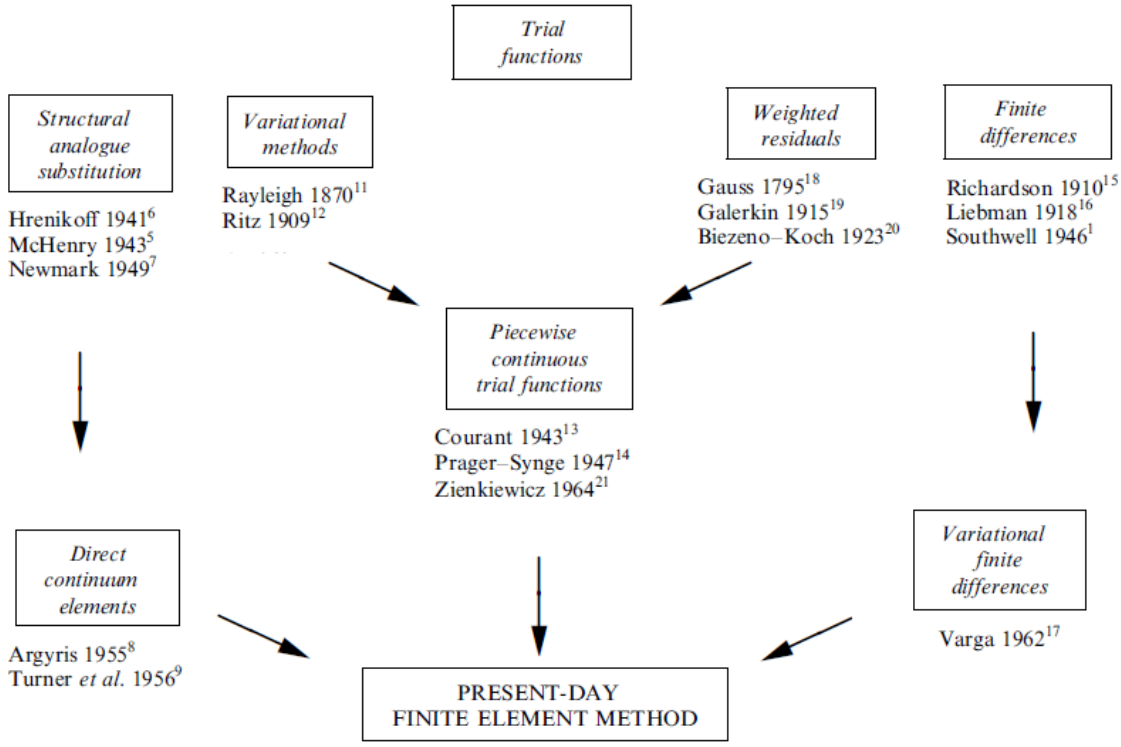

Figure 2.2: Historical Evolution of the FEM [OCZ00].

finite element process states that: "the continuum is divided into a finite number of parts (elements), the behaviour of which is specified by a finite number of parameters, and the solution of the complete system as an assembly of its elements follows precisely the same rules as those applicable to standard discrete problems".

This methodology was being developed side by side with the digital computer and they complement each other. This means that to achieve better results with FEM the number of elements should be studied and refined and the use of computers reduces the calculation time. Otherwise

the most precise analysis would take so much time that it would be impractical to solve certain problems. One of the first companies that developed software which used finite element analysis (FEA) was ANSYS in 1970.

To perform a FEA it is required to follow a methodology, summarized in Figure 2.3, before achieving the final solution. First, it is necessary to look at the physical problem and define the mathematical differential equations to solve, including the boundary and initial conditions. Then, the domain needs to be split to define the elements and its nodes and create the so-called mesh. After doing this, it is possible to solve the DE using FEM, although it is then necessary to look at the results and check if they are plausible. If not, the mathematical model needs to be verified to check if some physical incoherence is present or if the mesh is incorrectly defined. If the outputs are physically plausible, a mesh refinement can be done to achieve better accuracy. This last step needs to be performed considering the hardware properties, because with a very detailed mathematical model or with many elements and nodes (or both) a large amount of computer memory and processor capacity is required.

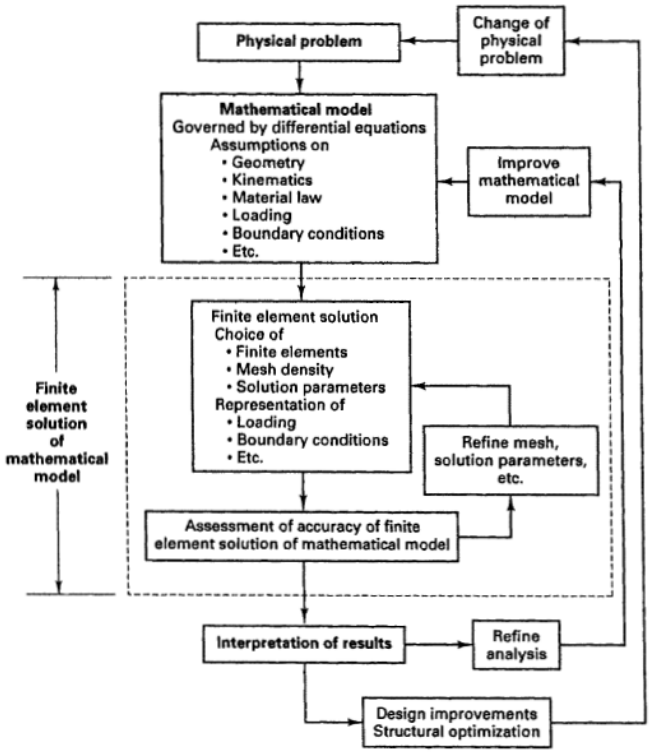

Figure 2.3: Algorithm for a FEM Analysis [Bat14].

In the present days FEA is strongly related with computer aided design (CAD) and because of that one other step can be added to this methodology that is the graphical representation of the results. This helps the user to check if the process was done correctly because he has a clear image of the entire process.

2.2 Constitutive model

In structural mechanics the constitutive model is no more than an equation that establishes the relation between stresses and strains. The strains are linear and angular displacements in a deformable body [Meg07]. They are the quotient between the deformation suffered and the initial length of a small element within the structure. So, it is first required to define a vector 2.1 that represent the spacial displacement that can occur.

$$s(x,y,z,t) = \begin{bmatrix} u(x,y,z,t) \\ v(x,y,z,t) \\ w(x,y,z,t) \end{bmatrix} \quad (2.1)$$

This vector is filled with the displacements u, v and w in the three space directions x, y and z respectively. These displacements are time dependent, once they can vary in time depending on the loads applied. However, the work developed in this thesis ignores the time dependence of the displacements. In other words, this is a static analysis where just the space variables are required 2.2.

$$\overline{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.2)$$

It is also necessary to define a unit vector that is associated with the orientation of each axis and those are defined by the letters i, j and k for the x-, y- and z-axis respectively. It is then possible to calculate the strains 2.3 along each axis and in the different planes.

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial s_i}{\partial \overline{X_j}} + \frac{\partial s_j}{\partial \overline{X_i}}\right) \quad (2.3)$$

As it is possible to check in 2.4 in the main diagonal of the matrix there are represented the strains along each axis - normal strain - detonated by ε (sub axis) and the strains in the planes - shear strain - detonated by γ (sub plane). This is a symmetric matrix therefore it is equivalent to the final column matrix of 2.5.

$$\overline{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) & \frac{1}{2}(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}) \\ \frac{1}{2}(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) & \frac{\partial v}{\partial y} & \frac{1}{2}(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}) \\ \frac{1}{2}(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) & \frac{1}{2}(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}) & \frac{\partial w}{\partial z} \end{bmatrix} \quad (2.4)$$

$$\widehat{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\gamma_{xy} \\ 2\gamma_{xz} \\ 2\gamma_{yz} \end{bmatrix} \quad (2.5)$$

So, to have coherence in the constitutive model, the constitutive matrix 2.6 is a six by six matrix. Its content is provided by studying the material and its properties, so it is entirely dependent of the material that is being analysed and strongly affects the final results of the structural analysis.

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \quad (2.6)$$

Finally, it is possible to determine the stress vector using Hooke's law 2.8. This law states that the Cauchy stress $\widehat{\sigma}$ tensor 2.9 is the result of the product of stiffness tensor (constitutive matrix) and strain tensor (assumed as strain vector for simplicity purposes). The normal stress is detonated by σ (sub axis) and the shear stress by τ (sub plane).

$$\overline{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad (2.7)$$

$$\widehat{\sigma} = C\widehat{\varepsilon} \quad (2.8)$$

$$\widehat{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{bmatrix} \quad (2.9)$$

In this work, it is assumed that the material is isotropic, so the constitutive matrix is a diagonal

matrix 2.10. In this matrix the three first diagonal terms are the Young's modulus (E) and are related with the normal strains and the other three are shear modulus (G) that are related with the shear strains. The relationship between those two constants is given by Equation 2.11 using the Poisson's ratio (ν).

$$C = \begin{bmatrix} E & 0 & 0 & 0 & 0 & 0 \\ 0 & E & 0 & 0 & 0 & 0 \\ 0 & 0 & E & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad (2.10)$$

$$G = \frac{E}{2(1+\nu)} \quad (2.11)$$

As already enunciated above, the structure is studied taking into account three different sub problems that are divided accordingly the loads and displacements orientation. This three problems are described in the following chapter's sub-sections.

2.3 Bar Problem

The first case is the bar element which withstands only axial loads, represented in Figure 2.4, and undergoes only axial displacements. Those loads can be concentrated on any of the two extremities of the bar element, P_x, or may be distributed along its length, Q_x.

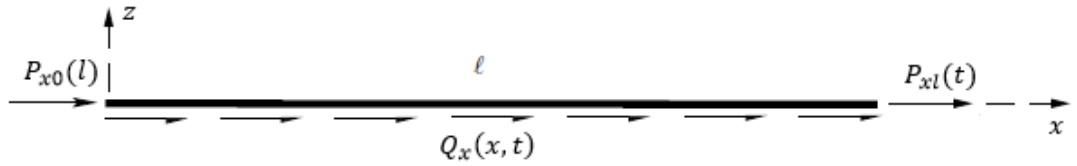

Figure 2.4: Bar Loads

The first step is to reduce the space displacements vector just to the axial displacement along x-axis 2.12. Then the strain 2.13 and stress 2.14 vectors are obtained from equations 2.5 and 2.9 respectively.

$$s(x,t) = u(x,t) \quad (2.12)$$

$$\varepsilon = \varepsilon_{xx} = \varepsilon_x = \frac{\partial u}{\partial x} \quad (2.13)$$

$$\sigma = \sigma_{xx} = \sigma_x = C_{11}\varepsilon_x = E\frac{\partial u}{\partial x} \quad (2.14)$$

The next step is to determine the dynamic equilibrium equations. There are several ways to achieve this:

1. Newton's second law;
2. D'Alembert's principle;
3. Energetic formulation;
4. Principle of virtual work (PVW);
5. General dynamics equation;
6. Lagrange equation.

For this thesis the principle chosen was the principle of virtual work because it easily enables to introduce the virtual variations that will be hereafter replaced by the approximation functions.

So, the variation of the internal work δW_{int} is determined by Equation 2.15 and it is equal to the variation on the stress within the body volume dV.

$$\delta W_{int} = \int_V \delta\varepsilon_x \sigma_x dV \quad (2.15)$$

Replacing θ_x and σ_x by 2.13 and 2.14 the variation of internal work is now represented by Equation 2.16.

$$\delta W_{int} = \int_V \delta\frac{\partial u}{\partial x} E\frac{\partial u}{\partial x} dV \quad (2.16)$$

This volume integral is then decomposed in an area integral where the cross-section properties of the structure are present and a length variation that gives us the 1D shape for the FEM. It is assumed that the material properties do not change across the cross-section area, so it is possible to assume EA as the rod axial stiffness and obtain 2.17.

$$\delta W_{int} = \int_l \delta\frac{\partial u}{\partial x} EA\frac{\partial u}{\partial x} dx \quad (2.17)$$

On the other side of the principle of virtual work is the variation of the external work δW_{ext}. The variation work of inertia forces $\delta W_{ext_{inertia}}$ and the variation work of the applied loads $\delta W_{ext_{loads}}$ are here represented 2.18.

$$\delta W_{ext} = \delta W_{ext_{inertia}} + \delta W_{ext_{loads}} \quad (2.18)$$

Analysing the loads applied to the bar element (Figure 2.4) the variation of the external work is determined using Equation 2.19. The inertia load is the product of the displacement s and the bar density ρ.

$$\delta W_{ext} = -\int_V \delta s \rho \ddot{s} dV + \int_l \delta s Q_x dx + \delta u(0,t) P_{x0} + \delta u(l,t) P_{xl} \quad (2.19)$$

In Equation 2.19 the general displacement vector s can be replaced by the respective displacement for this case as Equation 2.12 shows. And once again, the volume integral is decomposed to obtain a length integral where it is possible to apply the FEM. It is also assumed that the material properties do not change across the section's area it is possible to assume m as the rod's mass per unit of length (= ρ A). The final Equation 2.20 for the variation of the external work is then obtained.

$$\delta W_{ext} = -\int_l \delta u m \ddot{u} dx + \int_l \delta u Q_x dx + \delta u(0,t) P_{x0} + \delta u(l,t) P_{xl} \quad (2.20)$$

The next required step is to obtain the approximation functions for the displacement that will be applied in the PVW. Let us assume the discretization of the position domain looking for a single element with two nodes 2.5.

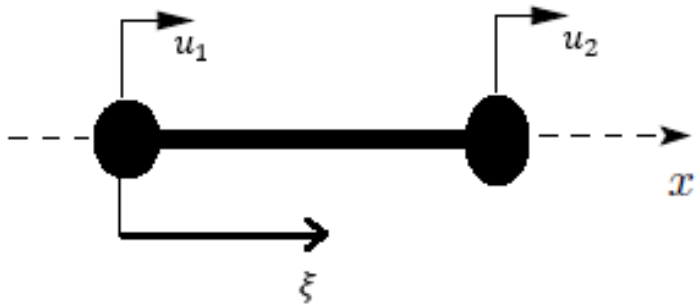

Figure 2.5: Bar Displacements

Let us introduce some variables that are illustrated in the Figure 2.5:

ξ - position along the element

u_{j-1}(t) - displacement of the element left node

u_j(t) - displacement of the element right node

l_e - length of the element

u_{j-1} is at ξ = 0 and u_j is at ξ = 1

Now it is required to have the relation between ξ and x 2.21 because the FEM approximation considers a local displacement instead of global one. So, ξ is the local space variable for each element and it is dependent on the global position and the element length. It is also necessary to have the relation between this new local variable and the variation of the global position x of the element 2.22.

$$\xi = \frac{x - (j-1)l_e}{l_e} \quad (2.21)$$

$$dx = l_e d\xi \quad (2.22)$$

After understanding this changing of variables, a linear interpolation is applied to interpolate the element's displacement between the two nodes 2.23.

$$u(\xi, t) = a + b\xi \quad (2.23)$$

And with the boundary conditions 2.24 it is possible to determine the constants a 2.25.

$$\begin{aligned} u(0,t) &= u_1 \\ u(1,t) &= u_2 \end{aligned} \quad (2.24)$$

$$\begin{aligned} a &= u_1 \\ b &= u_2 - u_1 \end{aligned} \quad (2.25)$$

The final approximation function for de displacement u 2.26 is obtain by replacing the constants a and b in 2.23 with 2.25.

$$u(\xi, t) = (1 - \xi)u_2 + \xi u_j \quad (2.26)$$

Equation 2.26 can be written in a vector form 2.27 assuming a vector for the nodes displacement 2.28 and one for the local position variable 2.29.

$$u(\xi, t) = N(\xi) d_e(t) \quad \textbf{(2.27)}$$

$$d_e(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad \textbf{(2.28)}$$

$$N(\xi) = \begin{bmatrix} 1 - \xi & \xi \end{bmatrix} \quad \textbf{(2.29)}$$

$$\frac{dN}{d\xi} = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad \textbf{(2.30)}$$

And lastly, analysing the PVW equations, the variation of this approximation function 2.31 it is also required.

$$\delta u(\xi, t) = N(\xi) \delta d_{ej}(t) \quad \textbf{(2.31)}$$

Going back to the PVW the internal and external work must be equal 2.32 and 2.33.

$$\delta W_{int} = \delta W_{ext} \quad \textbf{(2.32)}$$

$$\int_l \delta \frac{\partial u}{\partial x} EA \frac{\partial u}{\partial x} dx = -\int_l \delta u m \ddot{u} dx + \int_l \delta u Q_x dx + \delta u(0,t) P_{x0} + \delta u(l,t) P_{xl} \quad \textbf{(2.33)}$$

Since vectors and matrices in Equation 2.33 require some algebraic multiplication it is necessary to transpose the first displacement in each equation's term. This will lead to Equation 2.34.

$$\int_l \delta \frac{\partial u}{\partial x}^T EA \frac{\partial u}{\partial x} = -\int_l \delta u^T m \ddot{u} dx + \int_l \delta u^T Q_x dx + \delta u^T(0,t) P_{x0} + \delta u^T(l,t) P_{xl} \quad \textbf{(2.34)}$$

Finally, it is possible to perform the coordinate change 2.35 and insert the approximation function and starting by consider just one element the final FEM equation for this problem is achieved 2.37.

$$\int_0^1 \delta \frac{\partial (Nd_e)^T}{\partial \xi l_e} EA_e \frac{\partial (Nd_e)}{\partial \xi l_e} l_e d\xi = \\ -\int_0^1 (N\delta d_e)^T m_e N \ddot{d_e} l_e d\xi + \int_0^1 (N\delta d_e)^T Q_{x_e} l_e d\xi + (N(0)\delta d_e)^T P_{x1} + (N(1)\delta d_e)^T P_{x2} \quad (2.35)$$

$$\delta d_e^T \frac{1}{l_e} \int_0^1 \frac{\partial N^T}{\partial \xi} EA_e \frac{\partial N}{\partial \xi} d\xi d_e = \\ -\delta d_e^T l_e \int_0^1 N^T m_e N d\xi \ddot{d_e} + \delta (d_e)^T l_e \int_0^1 N^T Q_{x_e} d\xi + \delta d_e^T N(0)^T P_{x1} + \delta d_e^T N(1)^T P_{x2} \quad (2.36)$$

$$\delta d_e^T K_e d_e = -\delta d_e^T M_e \ddot{d_e} + \delta d_e^T F_e \quad (2.37)$$

In these final equations K_e represents the stiffness matrix 2.38, M_e represents the mass matrix 2.39, F_e represents the load vector 2.40 and d_e the displacements vector 2.44 for one element.

$$K_e = \frac{1}{l_e} \int_0^1 \frac{\partial N^T}{\partial \xi} EA_e \frac{\partial N}{\partial \xi} d\xi \quad (2.38)$$

$$M_e = l_e \int_0^1 N^T m_e N d\xi \quad (2.39)$$

$$F_e = l_e \int_0^1 N^T Q_{x_e} d\xi + N^T(0) P_{x1} + N^T(1) P_{x2} \quad (2.40)$$

Replacing in this last three equations the vector N and its variation by the Equations 2.29 and 2.30 the final matrices for the elements are:

$$K_e = \frac{EA_e}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.41)$$

$$M_e \frac{l_e m_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (2.42)$$

$$F_e = \frac{l_e Q_{x_e}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + P_{x1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + P_{x2} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.43)$$

$$d_e = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2.44)$$

Up to this point it was considered just one element, although the finite element method is the assembly of several elements so it is required to perform the sum of the PVW of each element 2.45, where N is the number of elements and now the properties and loads are functions of the local coordinate ξ.

$$\begin{aligned} \sum_{j=1}^{N} \int_{(j-1)l_e}^{jl_e} \delta \frac{\partial u^T}{\partial x} EA_{ej}(\xi) \frac{\partial u}{\partial x} dx = \\ -\sum_{j=1}^{N} \int_{(j-1)l_e}^{jl_e} \delta u^T m_{ej}(\xi) \ddot{u} dx + \sum_{j=1}^{N} \int_{(j-1)l_e}^{jl_e} \delta u^T Q_{x_{ej}}(\xi) dx + \delta u^T(0,t) P_{x1} + \delta u^T(l,t) P_{x2} \end{aligned} \quad (2.45)$$

A summation of Equation 2.37 is required for all elements that leads to Equation 2.46 in which K_{ej}, M_{ej}, F_{ej}, d_{ej} are the stiffness and mass matrices and loads and displacements vector, respectively, of element j.

$$\sum_{j=1}^{N} \delta d_{ej}^T K_{ej} d_{ej} = -\sum_{j=1}^{N} \delta d_{ej}^T M_{ej} \ddot{d_{ej}} + \sum_{j=1}^{N} \delta d_{ej}^T F_{ej} \quad (2.46)$$

In a practical computational application, a final matrix is assembled as shown in Figure 2.6 where the result is an n vs. n matrix for the stiffness and mass matrices and a n column vector for the loads and displacements vectors, where n is the number of degrees of freedom (DOF) of the problem. For 1D linear elements the number of nodes is equal to two times the number of elements plus one.

2.4 Rod Problem

The formulation of the finite element method for a rod is similar to the bar, therefore the different steps are here demonstrated.

In Figure 2.7 it is possible to see the loads present in the rod formulation. This loads are a distributed torsion moment W_x and two concentrated torsion moments M_x one at each rod's tip.

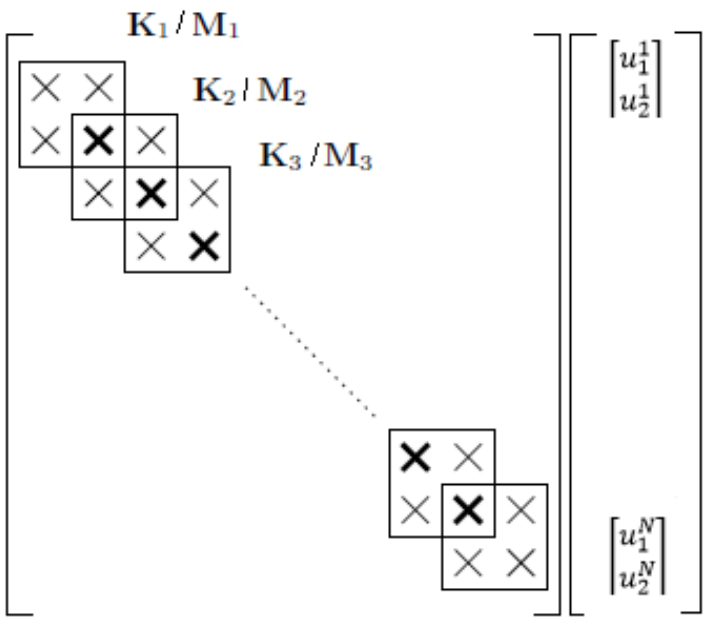

Figure 2.6: Bar Matrices

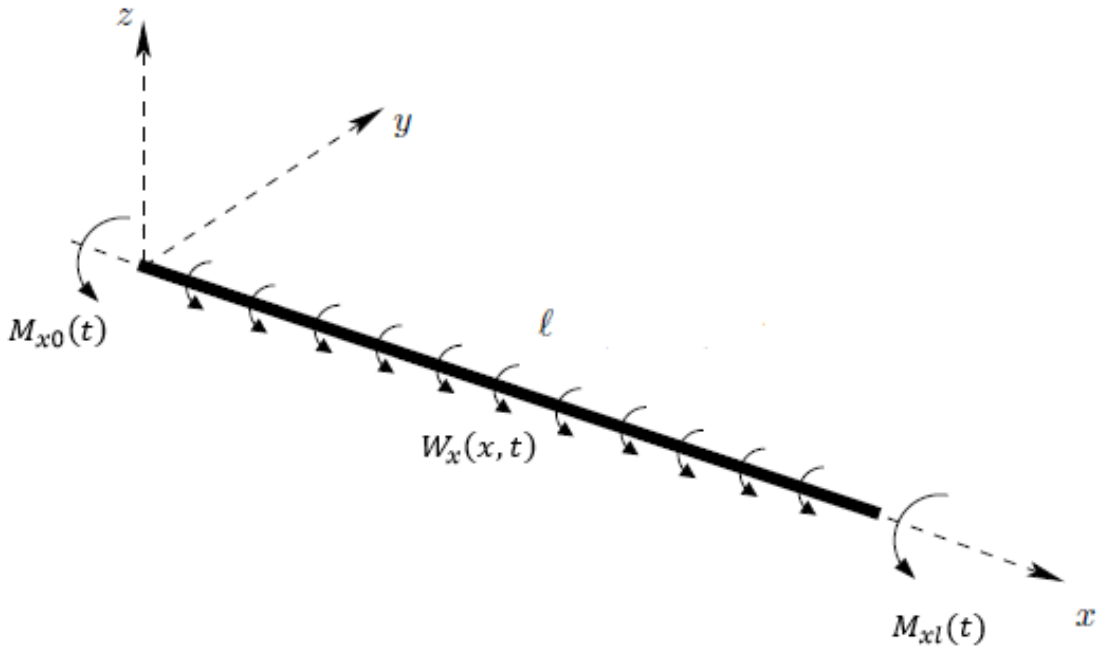

Figure 2.7: Rod Loads

The displacement in this case is the torsion along the x axis θ_x and the displacements vector is now represented as shown in Equation 2.47. The ψ function is the contribution of the cross-section warping.

$$s(x,y,z,t) = \begin{bmatrix} \psi(y,z)\frac{\partial \theta_x}{\partial x} \\ -z\theta_x \\ y\theta_x \end{bmatrix} \quad (2.47)$$

The strain and stress vectors are obtained by substituting the displacement in Equation 2.3.

$$\varepsilon = \begin{bmatrix} 0 \\ 0 \\ 0 \\ (\frac{\partial \psi}{\partial y} - z)\frac{\partial \theta_x}{\partial x} \\ (\frac{\partial \psi}{\partial z} + y)\frac{\partial \theta_x}{\partial x} \\ 0 \end{bmatrix} \quad (2.48)$$

$$\sigma = \begin{bmatrix} 0 \\ 0 \\ 0 \\ G(\frac{\partial \psi}{\partial y} - z)\frac{\partial \theta_x}{\partial x} \\ G(\frac{\partial \psi}{\partial z} + y)\frac{\partial \theta_x}{\partial x} \\ 0 \end{bmatrix} \quad (2.49)$$

The variation of the internal work δW_{int} is given by equation 2.50 and replacing ε and θ by the values in Equations 2.48 and 2.49 it is possible to obtain Equation 2.51.

$$\delta W_{int} = \int_V \delta \varepsilon \sigma dV \quad (2.50)$$

$$\delta W_{int} = \int_V \delta \left(\left(\frac{\partial \psi}{\partial y} - z \right) \frac{\partial \theta_x}{\partial x} \right) G \left(\frac{\partial \psi}{\partial y} - z \right) \frac{\partial \theta_x}{\partial x} + \delta \left(\left(\frac{\partial \psi}{\partial z} + y \right) \frac{\partial \theta_x}{\partial x} \right) G \left(\frac{\partial \psi}{\partial z} + y \right) \frac{\partial \theta_x}{\partial x} dV \quad (2.51)$$

The volume integral is then separated into two other integrals for the area and length and assuming that the material properties do not change across the cross-section area it is possible to assume GJ as the bar torsional stiffness and the variation of the internal work is now:

$$\delta W_{int} = \int_l \delta \frac{\partial \theta_x}{\partial x} GJ \frac{\partial \theta_x}{\partial x} dx \quad (2.52)$$

Once again the variation of the external work δW_{ext} is the sum of the inertia and loads works as represented in the following equation:

$$\delta W_{ext} = \delta W_{ext_{inertia}} + \delta W_{ext_{loads}} \quad (2.53)$$

The last two works can be replaced by its terms and Equation 2.54 is achieved.

$$\delta W_{ext} = -\int_V \delta s \rho \ddot{s} dV + \int_l \delta s W_x dx + \delta\theta_x(0,t) M_{x0} + \delta\theta_x(l,t) M_{xl} \quad \textbf{(2.54)}$$

Replacing in Equation 2.54 the displacement by 2.47 and after performing some algebra calculations the variation of the external work for this case is given by Equation 2.55. In this equation I_p represents the inertial polar moment. Other constant I_ψ related to the warping of the cross-section was derived in this step, but it was neglected because its dimension is very small when compared with the polar moment of inertia.

$$\delta W_{ext} = -\int_l \delta\theta_x I_p \ddot{\theta_x} dx + \int_l \delta\theta_x W_x dx + \delta\theta_x(0,t) M_{x0} + \delta\theta_x(l,t) M_{xl} \quad \textbf{(2.55)}$$

Then it is required to obtain the approximation function for the displacement θ_x. This step is the same as the bar.

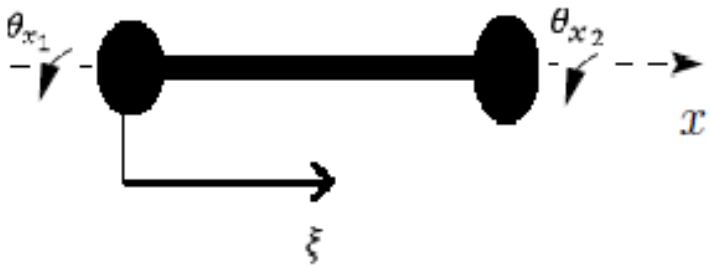

Figure 2.8: Rod Displacements

ξ - position along the element

$\theta_{x_{j-1}}$(t) - displacement of the element left node

θ_{xj}(t) - displacement of the element right node

l_e - length of the element

$\theta_{x_{j-1}}$ is at ξ = 0 and θ_{xj} is at ξ = 1

So the final equations are the following:

$$d_e(t) = \begin{bmatrix} \theta_{x_1}(t) \\ \theta_{x_2}(t) \end{bmatrix} \quad \textbf{(2.56)}$$

$$N(\xi) = \begin{bmatrix} 1-\xi & \xi \end{bmatrix} \quad \textbf{(2.57)}$$

$$\frac{dN}{d\xi} = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad (2.58)$$

$$\delta\theta_x(\xi, t) = N(\xi)\delta d_{ej}(t) \quad (2.59)$$

Going back to the PVW and replacing in Equation 2.63 by the values on the Equations 2.52 and 2.55 and transposing the matrices to allow the algebraic computations the Equation 2.61 is obtained.

$$\delta W_{int} = \delta W_{ext} \quad (2.60)$$

$$\int_l \delta \frac{\partial \theta_x}{\partial x}^T GJ \frac{\partial \theta_x}{\partial x} = -\int_l \delta\theta_x^T I_p \ddot{\theta_x} dx + \int_l \delta\theta_x^T W_x dx + \delta\theta_x^T(0,t)M_{x0} + \delta\theta_x^T(l,t)M_{xl} \quad (2.61)$$

After performing the coordinate change to the local coordinate ξ and applying the approximation function 2.59 in the motion equation 2.61 it is possible to have the FEM equation for the rod 2.62, where it is assumed that GJ and I_p do not change along the element.

$$\begin{aligned} \delta d_e^T \frac{1}{l_e} \int_0^1 \frac{\partial N^T}{\partial \xi} GJ_e \frac{\partial N}{\partial \xi} d\xi d_e = \\ -\delta d_e^T l_e \int_0^1 N^T I_{p_e} N d\xi \ddot{d}_e + \delta(d_e)^T l_e \int_0^1 N^T W_{x_e} d\xi + \delta d_e^T N(0)^T M_{x1} + \delta d_e^T N(1)^T M_{x2} \end{aligned} \quad (2.62)$$

The last equation 2.62 can be simplified to Equation 2.63.

$$\delta d_e^T K_e d_e = -\delta d_e^T M_e \ddot{d}_e + \delta d_e^T F_e \quad (2.63)$$

In which K_e, M_e, F_e and d_e represent the stiffness 2.64 and mass 2.65 matrices and loads 2.66 and displacements 2.67 vectors respectively.

$$K_e = \frac{GJ_e}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.64)$$

$$M_e = \frac{l_e I_{p_e}}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (2.65)$$

$$F_e = \frac{l_e W_{x_e}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + M_{x1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + M_{x2} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.66)$$

$$d_e = \begin{bmatrix} \theta_{x_1} \\ \theta_{x_2} \end{bmatrix} \quad (2.67)$$

2.5 Beam 1 Problem

The general formulation of the beam problem is equal to the previous ones, the only main difference is that in this problem there are two displacements to take into account. This two displacements are a linear displacement w and a rotation θ_x.

By looking to Figure 2.9 the present loads in this case are:

1. distributed load aligned with z-axis Q_z;
2. distributed bending moment W_y;
3. concentrated loads aligned with z-axis P_z, one at each beam tip;
4. concentrated bending moment M_y, one at each beam tip.

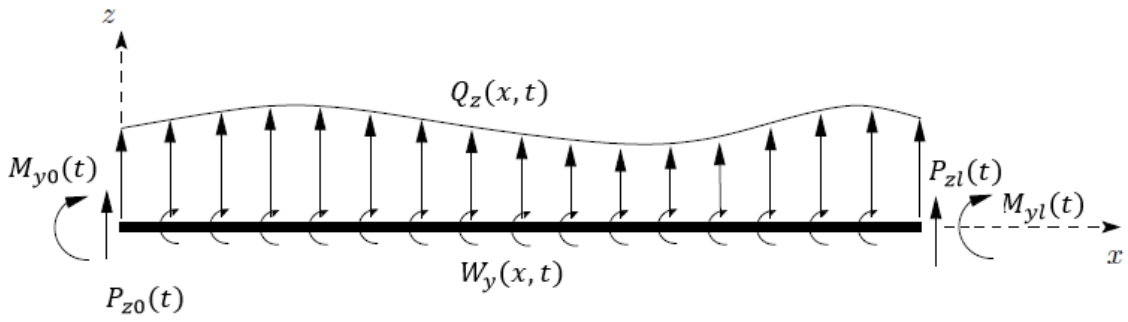

Figure 2.9: Beam 1 Loads

To simplify this case it is possible to assume a decoupled Euler-Bernoulli beam theory so the displacements vector is represented by equation 2.68.

$$s(x,t) = \begin{bmatrix} -z\frac{\partial w}{\partial x} \\ 0 \\ 0 \end{bmatrix} \quad (2.68)$$

$$\theta_y = \frac{\partial w}{\partial x} \quad (2.69)$$

Assuming the relation of the previous equation 2.69 and using Equation 2.3 it is possible to achieve the strain vector 2.70 for the beam.

$$\varepsilon = \begin{bmatrix} -z\frac{\partial^2 w}{\partial x^2} \\ 0 \\ 0 \end{bmatrix} \quad (2.70)$$

Once having the strain vector, using the constitutive relation the stress vector is equal to 2.71.

$$\sigma = E \begin{bmatrix} -z\frac{\partial^2 w}{\partial x^2} \\ 0 \\ 0 \end{bmatrix} \quad (2.71)$$

After this first step, the second one is to calculate motion equation for the beam. To perform this step the PVW is used and the variation of the internal work δW_{int} is given by the following equation:

$$\delta W_{int} = \int_V \delta\varepsilon\sigma dV \quad (2.72)$$

Using the Equations 2.70 and 2.71 in Equation 2.72, Equation 2.73 is obtained.

$$\delta W_{int} = \int_V \delta\frac{\partial^2 w}{\partial x^2} z^2 E \frac{\partial^2 w}{\partial x^2} dV \quad (2.73)$$

In the last equation the volume integral is transformed into an area and a length integrals and assuming that the material properties do not change across the cross-section area it is possible to assume EI_{yy} as the beam bending stiffness. After integrate the equation by parts twice and assuming EI_{yy} constant along x-axis the variation of the internal work for this beam is given by Equation 2.74.

$$\delta W_{int} = \left[\delta\frac{\partial w}{\partial x} EI_{yy}\frac{\partial^2 w}{\partial x^2}\right]_0^l - \left[\delta w EI_{yy}\frac{\partial^3 w}{\partial x^3}\right]_0^l + \int_l \delta w EI_{yy}\frac{\partial^4 w}{\partial x^4} dx \quad (2.74)$$

Then it is required to have the variation of the external work δW_{ext}. This is the sum 2.75 of the

work done by the beam inertia and the external loads applied as Figure 2.9 shows. The inertial loads are related with the displacement s and the beam density ρ.

$$\delta W_{ext} = \delta W_{ext_{inertia}} + \delta W_{ext_{loads}} \quad (2.75)$$

The Equation 2.75 is then rewritten as Equation 2.76.

$$\delta W_{ext} = -\int_V \delta s \rho \ddot{s} dV + \int_l \delta w Q_z dx \int_l \delta \frac{\partial w}{\partial x} W_y dx + \delta w(0,t) P_{z0} + \delta w(l,t) P_{zl} + \delta \frac{\partial w}{\partial x}(0,t) M_{y0} + \delta \frac{\partial w}{\partial x}(l,t) M_{yl} \quad (2.76)$$

Once again it is assumed that the material properties do not change across the cross-section area and remember the Equation 2.69 the variation of the external work is:

$$\delta W_{ext} = -\int_l \delta w m \ddot{w} dx + \int_l \delta w Q_z dx + \int_l \delta \theta_y W_y dx + \delta w(0,t) P_{z0} + \delta w(l,t) P_{zl} + \delta \theta_y(0,t) M_{y0} + \delta \theta_y(l,t) M_{yl} \quad (2.77)$$

Then it is necessary to have the approximation function for the displacements. Like the previous cases lets start by assuming a variable change. It is assumed the discretization of the position domain looking for a single element with two nodes, as represented in 2.10.

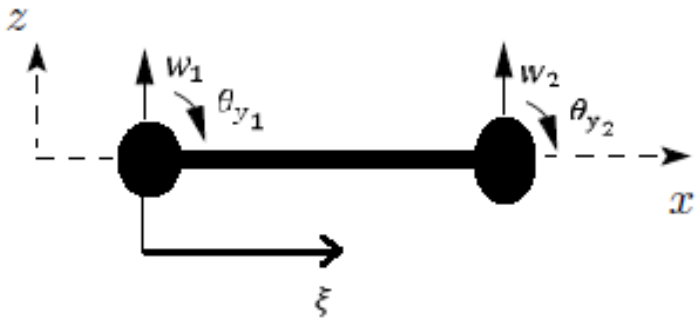

Figure 2.10: Beam 1 Displacements

And it is also required to assume some variables:

ξ - position along the element

w_{j-1}(t) - displacement of the element left node

w_j(t) - displacement of the element right node

$\theta_{y_{j-1}}$(t) - displacement of the element left node

θ_{y_j}(t) - displacement of the element right node

l_e - length of the element

w_{j-1} is at ξ = 0 and w_j is at ξ = 1

$\theta_{y_{j-1}}$ is at ξ = 0 and θ_{y_j} is at ξ = 1

The first step is to calculate the relation between ξ and x, that is given by Equation 2.78 and the variation by 2.79.

$$\xi = \frac{x - (j-1)l_e}{l_e} \quad (2.78)$$

$$dx = l_e d\xi \quad (2.79)$$

Similar to the bar and rod cases a linear interpolation is used in order to interpolate the element displacement between the two nodes. Although, for the beam problem there are four displacements to take into account, so a fourth order polynomial function is used 2.80.

$$w(\xi, t) = a + b\xi + c\xi^2 + d\xi^3 \quad (2.80)$$

And using the boundary conditions $w(0,t)$ and $w(1,t)$ it is possible to have the relations 2.81 and 2.82.

$$w_1 = a \quad (2.81)$$

$$w_2 = a + b + c + d \quad (2.82)$$

And perform the derivative of equation 2.80:

$$\frac{dw}{d\xi} = b + 2c\xi + 3d\xi^2 \quad (2.83)$$

Remember the relation between the vertical displacement w and the θ_y for the first node, where the local coordinate ξ is zero:

$$\frac{dw}{dx}(0) = \theta_{y_1} \quad (2.84)$$

And using the equation 2.79 it is possible to have the derivative of the displacement w in function of the local coordinate ξ 2.85.

$$\frac{dw}{d\xi}(0) = l_e \frac{dw}{dx}(0) \quad (2.85)$$

After combining Equations 2.84 and 2.85, Equation 2.86 is reached:

$$\frac{dw}{d\xi}(0) = l_e \theta_{y_1} \quad (2.86)$$

And finally applying Equation 2.83 for the first node where the local coordinate is zero it is possible to have the relation 2.87.

$$l_e \theta_{y_1} = b \quad (2.87)$$

Using the same explanation as above for the second node it is possible to have the relation 2.88.

$$l_e \theta_{y_1} = b + 2c + 3d \quad (2.88)$$

It is now possible to combine the Equations: 2.81, 2.82 2.87 and 2.88 to have an equation system 2.89.

$$\begin{cases} w_1 = a \\ l_e \theta_{y_1} = b \\ w_2 = a + b + c + d \\ l_e \theta_{y_2} = b + 2c + 3d \end{cases} \quad (2.89)$$

$$\begin{cases} a = w_1 \\ b = l_e\theta_{y_1} \\ c = w_2 - w_1 - l_e\theta_{y_1} - d \\ 3d = l_e\theta_{y_2} - l_e\theta_{y_1} - 2(w_2 - w_1 - l_e\theta_{y_1} - d) \end{cases} \quad (2.90)$$

$$\begin{cases} a = w_1 \\ b = l_e\theta_{y_1} \\ c = w_2 - w_1 - l_e\theta_{y_1} - d \\ d = l_e\theta_{y_2} - l_e\theta_{y_1} - 2w_2 + 2w_1 + 2l_e\theta_{y_1} \end{cases} \quad (2.91)$$

$$\begin{cases} a = w_1 \\ b = l_e\theta_{y_1} \\ c = -3w_1 - 2l_e\theta_{y_1} + 3w_2 - l_e\theta_{y_2} \\ d = 2w_1 + l_e\theta_{y_1} - 2w_2 + l_e\theta_{y_2} \end{cases} \quad (2.92)$$

After solving the equation system it is possible to have the constants 2.92 of the approximation equation 2.80 and rewrite it as Equation 2.93.

$$w(\xi, t) = w_1(1 - 3\xi^2 + 2\xi^3) + l_e\theta_{y_1}(\xi - 2\xi^2 + \xi^3) + w_2(3\xi^2 - 2\xi^3) + l_e\theta_{y_2}(-\xi^2 + \xi^3) \quad (2.93)$$

The last equation can be compacted in a matrix form where N is the local coordinate matrix 2.96 and d_e the displacement matrix 2.95 for the two nodes of the element.

$$w(\xi, t) = N(\xi)d_e(t) \quad (2.94)$$

$$d_e(t) = \begin{bmatrix} w_1(t) \\ \theta_{y_1}(t) \\ w_2(t) \\ \theta_{y_2}(t) \end{bmatrix} \quad (2.95)$$

$$N(\xi) = \begin{bmatrix} 1 - 3\xi^2 + 2\xi^3 & l_e(\xi - 2\xi^2 + \xi^3) & 3\xi^2 - 2\xi^3 & l_e(-\xi^2 + \xi^3) \end{bmatrix} \quad (2.96)$$

Analysing the equations of the internal 2.74 and external 2.77 work it is required to have the first and second derivatives of the displacement function w in function of ξ. So these derivatives are represented in Equations 2.97 and 2.98.

$$\frac{dN}{d\xi} = \begin{bmatrix} -6\xi + 6\xi^2 & l_e(1 - 4\xi + 3\xi^2) & 6\xi - 6\xi^2 & l_e(-2\xi + 3\xi^2) \end{bmatrix} \quad \textbf{(2.97)}$$

$$\frac{d^2N}{d\xi^2} = \begin{bmatrix} -6 + 12\xi & l_e(-4 + 6\xi) & 6 - 12\xi & l_e(-2 + 6\xi) \end{bmatrix} \quad \textbf{(2.98)}$$

It is also necessary to have the variation of w:

$$\delta w(\xi, t) = N(\xi)\delta d_{ej}(t) \quad \textbf{(2.99)}$$

Matching the equation of the variation of internal work 2.74 with the equation of the variation of external work 2.77 it is possible to have Equation 2.100.

$$\begin{aligned} \int_l \delta \frac{\partial^2 w}{\partial x^2} EI_{yy} \frac{\partial^2 w}{\partial x^2} dx = \\ -\int_l \delta w m \ddot{w} dx + \int_l \delta w Q_z dx + \int_l \delta \frac{\partial w}{\partial x} W_y dx + \delta w(0,t) P_{z0} + \delta w(l,t) P_{zl} + \\ \delta \frac{\partial w}{\partial x}(0,t) M_{y0} + \delta \frac{\partial w}{\partial x}(l,t) M_{yl} \end{aligned} \quad \textbf{(2.100)}$$

Once again this equation uses matrices to perform the algebraic calculations correctly it is required to transpose the first matrix of each term.

$$\begin{aligned} \int_l \delta \frac{\partial^2 w}{\partial x^2}^T EI_{yy} \frac{\partial^2 w}{\partial x^2} dx = \\ -\int_l \delta w^T m \ddot{w} dx + \int_l \delta w^T Q_z dx + \int_l \delta \frac{\partial w}{\partial x}^T W_y dx + \delta w^T(0,t) P_{z0} + \delta w^T(l,t) P_{zl} + \\ \delta \frac{\partial w}{\partial x}^T (0,t) M_{y0} + \delta \frac{\partial w}{\partial x}^T (l,t) M_{yl} \end{aligned} \quad \textbf{(2.101)}$$

The finite element method is a sum of the PVW of each element so:

$$\sum_{j=1}^{N}\int_{(j-1)l_e}^{jl_e}\delta\frac{\partial^2 w}{\partial x^2}^T EJ_{y_{ej}}(\xi)\frac{\partial^2 w}{\partial x^2}dx =$$
$$-\sum_{j=1}^{N}\int_{(j-1)l_e}^{jl_e}\delta w^T m_{ej}\ddot{w}dx+\sum_{j=1}^{N}\int_{(j-1)l_e}^{jl_e}\delta w^T Q_{z_{ej}}(\xi)dx+\sum_{j=1}^{N}\int_{(j-1)l_e}^{jl_e}\delta\frac{\partial w}{\partial x}^T W_{y_{ej}}(\xi)dx \quad (2.102)$$
$$+\delta w^T(0,t)P_{z1}+\delta w^T(l,t)P_{z2}+\delta\frac{\partial w}{\partial x}^T(0,t)M_{y1}+\delta\frac{\partial w}{\partial x}^T(l,t)M_{y2}$$

In Equation 2.102 N is the number of elements.

Considering just one element, the last step to formulate the FEM for a beam is to use the approximation Function 2.94 and its derivatives and variation in the PVW 2.101 and obtain the FEM Equation 2.103.

$$\delta d_e^T\frac{1}{l_e^3}\int_0^1\frac{\partial^2 N}{\partial\xi^2}^T EI_{yy_e}\frac{\partial^2 N}{\partial\xi^2}d\xi d_e =$$
$$-\delta d_e^T l_e\int_0^1 N^T m_e N d\xi\ddot{d}_e+\delta d_e^T l_e\int_0^1 N^T Q_{z_e}d\xi+\delta d_e^T\frac{1}{l_e}\int_0^1\frac{\partial N}{\partial\xi}^T W_{y_e}d\xi \quad (2.103)$$
$$+\delta d_e^T N(0)^T P_{z1}+\delta d_e^T N(1)^T P_{z2}+\delta d_e^T\frac{\partial N}{\partial\xi l_e}^T(0)M_{y_1}+\delta d_e^T\frac{\partial N}{\partial\xi l_e}^T(1)M_{y_2}$$

The Equation 2.103 can be write in a compact form 2.104.

$$\delta d_e^T K_e d_e=-\delta d_e^T M_e\ddot{d}_e+\delta d_e^T F_e \quad (2.104)$$

In this compact form K_e 2.106, M_e 2.108, F_e 2.110 and d_e 2.111 are the stiffness and mass matrices and loads and displacements vectors respectively. This matrices are obtained after substituting the vector N 2.96 and its derivatives in Equations 2.105, 2.107 and 2.109.

$$K_e=\frac{1}{l_e^3}\int_0^1\frac{\partial^2 N}{\partial\xi^2}^T EI_{yy_e}\frac{\partial^2 N}{\partial\xi^2}d\xi \quad (2.105)$$

$$K_e=\frac{EIyy_e}{l_e^3}\int_0^1\begin{bmatrix}12 & 6l_e & -12 & 6l_e\\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2\\ -12 & -6l_e & 12 & -6l_e\\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2\end{bmatrix} \quad (2.106)$$

$$M_e = l_e \int_0^1 N^T m_e N d\xi \quad \text{(2.107)}$$

$$M_e = \frac{l_e m_e}{420} \begin{bmatrix} 156 & 22l_{ej} & 54 & -13l_{ej} \\ 22l_{ej} & 4l_{ej}^2 & 13l_{ej} & -3l_{ej}^2 \\ 54 & 13l_{ej} & 156 & -22l_{ej} \\ -13l_{ej} & -3l_{ej}^2 & -22l_{ej} & 4l_{ej}^2 \end{bmatrix} \quad \text{(2.108)}$$

$$\begin{aligned} F_e = l_e \int_0^1 N^T Q_{z_e} d\xi + N^T(0) P_{z1} + N^T(1) P_{z2} + \\ \frac{1}{l_e} \int_0^1 \frac{\partial N}{\partial \xi}^T W_{y_e} d\xi + \frac{\partial N}{\partial \xi l_e}^T (0) M_{y1} + \frac{\partial N}{\partial \xi l_e}^T (1) M_{y2} \end{aligned} \quad \text{(2.109)}$$

$$F_e = \frac{l_e Q_{z_e}}{2} \begin{bmatrix} 1 \\ \frac{l_e}{6} \\ 1 \\ -\frac{l_e}{6} \end{bmatrix} + P_{z1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + P_{z2} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{W_{y_e}}{l_e} \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{M_{y_1}}{l_e} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{M_{y_2}}{l_e} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{(2.110)}$$

$$d_e = \begin{bmatrix} w_1 \\ \theta_{y_1} \\ w_2 \\ \theta_{y_2} \end{bmatrix} \quad \text{(2.111)}$$

2.6 Beam 2 Problem

This beam is similar to the previous one, but the displacement that are implicit here are v and θ_z. Once the formulation is equal only the initial displacements 2.112, strain2.115 and stress vectors2.116 and the final matrices are displayed.

By looking to Figure 2.11 the present loads in this case are:

1. distributed load aligned with y-axis Q_y;
2. distributed bending moment W_z;
3. concentrated loads aligned with y-axis P_y, one at each beam tip;
4. concentrated bending moment M_z, one at each beam tip.

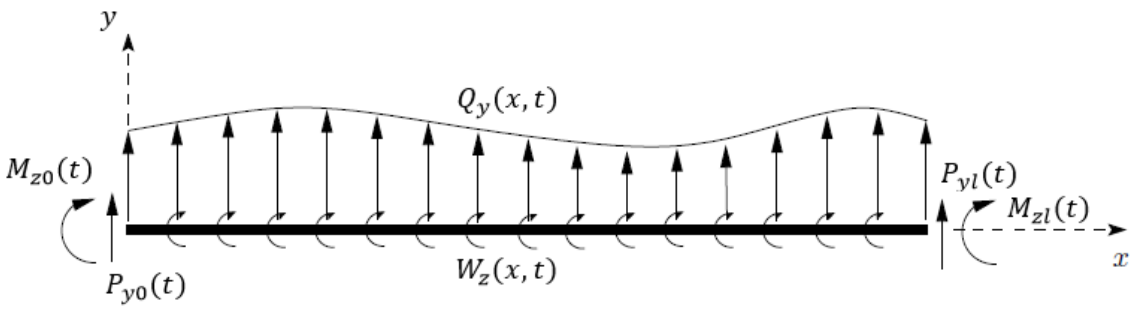

Figure 2.11: Beam 2

$$s(x,t) = \begin{bmatrix} y\theta_z(x,t) \\ v(x,t) \\ 0 \end{bmatrix} \quad (2.112)$$

It is considered that: $\gamma_{xy} = \gamma_{xz} = 0$ the cross section is small when compared with the length so:

$$\theta_z = -\frac{\partial v}{\partial x} \quad (2.113)$$

And replacing 2.113 in Equation 2.112 the displacements vector is:

$$s(x,t) = \begin{bmatrix} -y\frac{\partial v}{\partial x} \\ 0 \\ 0 \end{bmatrix} \quad (2.114)$$

$$\varepsilon = \begin{bmatrix} -y\frac{\partial^2 v}{\partial x^2} \\ 0 \\ 0 \end{bmatrix} \quad (2.115)$$

$$\sigma = C\varepsilon_x = E \begin{bmatrix} -y\frac{\partial^2 v}{\partial x^2} \\ 0 \\ 0 \end{bmatrix} \quad (2.116)$$

Assuming the discretization of the position domain and looking for a single element with two nodes 2.12 it is possible to obtain the approximation function.

ξ - position along the element

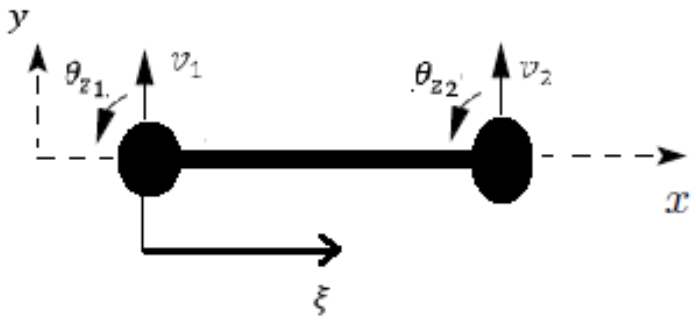

Figure 2.12: Beam 2 Displacements

v_{j-1}(t) - displacement of the element left node

v_j(t) - displacement of the element right node

$\theta_{z_{j-1}}$(t) - displacement of the element left node

θ_{z_j}(t) - displacement of the element right node

l_e - length of the element

v_{j-1} is at ξ = 0 and v_j is at ξ = 1

$\theta_{z_{j-1}}$ is at ξ = 0 and θ_{z_j} is at ξ = 1

$$v(\xi,t) = N(\xi)d_e(t) \quad \textbf{(2.117)}$$

$$d_e(t) = \begin{bmatrix} v_1(t) \\ \theta_{z_1}(t) \\ v_2(t) \\ \theta_{z_2}(t) \end{bmatrix} \quad \textbf{(2.118)}$$

$$N(\xi) = \begin{bmatrix} 1-3\xi^2+2\xi^3 & l_e(\xi-2\xi^2+\xi^3) & 3\xi^2-2\xi^3 & l_e(-\xi^2+\xi^3) \end{bmatrix} \quad \textbf{(2.119)}$$

$$\frac{dN}{d\xi} = \begin{bmatrix} -6\xi+6\xi^2 & l_e(1-4\xi+3\xi^2) & 6\xi-6\xi^2 & l_e(-2\xi+3\xi^2) \end{bmatrix} \quad \textbf{(2.120)}$$

$$\frac{d^2N}{d\xi^2} = \begin{bmatrix} -6+12\xi & l_e(-4+6\xi) & 6-12\xi & l_e(-2+6\xi) \end{bmatrix} \quad \textbf{(2.121)}$$

And finally:

$$\delta v(\xi,t) = N(\xi)\delta d_{ej}(t) \quad \textbf{(2.122)}$$

The final equation of motion is:

$$\begin{aligned} \delta d_e^T \frac{1}{l_e^3}\int_0^1 \frac{\partial^2 N}{\partial \xi^2}^T EI_{zz_e}\frac{\partial^2 N}{\partial \xi^2}d\xi d_e = \\ -\delta d_e^T l_e \int_0^1 N^T m_e N d\xi \ddot{d}_e + \delta d_e^T l_e \int_0^1 N^T Q_{y_e} d\xi + \delta d_e^T \frac{1}{l_e}\int_0^1 \frac{\partial N}{\partial \xi}^T W_{z_e} d\xi \\ +\delta d_e^T N(0)^T P_{y1} + \delta d_e^T N(1)^T P_{y2} + \delta d_e^T \frac{\partial N}{\partial \xi l_e}^T M_{z_1} + \delta d_e^T \frac{\partial N}{\partial \xi l_e}^T M_{z_2} \end{aligned} \quad \textbf{(2.123)}$$

Or in its compact form:

$$\delta d_e^T K_e d_e = -\delta d_e^T M_e \ddot{d}_e + \delta d_e^T F_e \quad \textbf{(2.124)}$$

$$K_e = \frac{EIzz_e}{l_e^3}\begin{bmatrix} 12 & 6l_e & -12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix} \quad \textbf{(2.125)}$$

$$M_e = \frac{l_e m_e}{420}\begin{bmatrix} 156 & 22l_{ej} & 54 & -13l_{ej} \\ 22l_{ej} & 4l_{ej}^2 & 13l_{ej} & -3l_{ej}^2 \\ 54 & 13l_{ej} & 156 & -22l_{ej} \\ -13l_{ej} & -3l_{ej}^2 & -22l_{ej} & 4l_{ej}^2 \end{bmatrix} \quad \textbf{(2.126)}$$

$$F_e = \frac{l_e Q_{y_e}}{2}\begin{bmatrix} 1 \\ \frac{l_e}{6} \\ 1 \\ -\frac{l_e}{6} \end{bmatrix} + P_{y1}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + P_{y2}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{W_{z_e}}{l_e}\begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{M_{z_1}}{l_e}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \frac{M_{z_2}}{l_e}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \textbf{(2.127)}$$

$$d_e = \begin{bmatrix} v_1 \\ \theta_{z_1} \\ v_2 \\ \theta_{z_2} \end{bmatrix} \quad \textbf{(2.128)}$$

2.7 General Problem

The last step is to join the four resultant matrices in one and it is possible to obtain the general matrices for one element with two tip nodes. The main requirement to do this matrix summation is to select an order for the displacements and then respect that order for the displacements, rigidity and loads matrices. The order chosen for this case is: beam 1, beam 2, rod and bar. That translates in terms of displacements as: w, v, u, θ_y, θ_z and θ_x for node one and two respectively.

$$d_e = \begin{bmatrix} w_1 \\ v_1 \\ u_1 \\ \theta_{y_1} \\ \theta_{z_1} \\ \theta_{x_1} \\ w_2 \\ v_2 \\ u_2 \\ \theta_{y_2} \\ \theta_{z_2} \\ \theta_{x_2} \end{bmatrix} \quad (2.129)$$

$$F_e = \begin{bmatrix} \frac{l_e Q_{z_e}}{2} + P_{z1} - \frac{W_{y_e}}{l_e} \\ \frac{l_e Q_{y_e}}{2} + P_{y1} - \frac{W_{z_e}}{l_e} \\ \frac{l_e Q_{x_e}}{2} + P_{x1} \\ \frac{l_e^2 Q_{z_e}}{12} + \frac{M_{y_1}}{l_e} \\ \frac{l_e^2 Q_{y_e}}{12} + \frac{M_{z_1}}{l_e} \\ \frac{l_e W_{x_e}}{2} + M_{x1} \\ \frac{l_e Q_{z_e}}{2} + P_{z2} + \frac{W_{y_e}}{l_e} \\ \frac{l_e Q_{y_e}}{2} + P_{y2} + \frac{W_{z_e}}{l_e} \\ \frac{l_e Q_{x_e}}{2} + P_{x2} \\ -\frac{l_e^2 Q_{z_e}}{12} + \frac{M_{y_2}}{l_e} \\ -\frac{l_e^2 Q_{y_e}}{12} + \frac{M_{z_2}}{l_e} \\ \frac{l_e W_{x_e}}{2} + M_{x2} \end{bmatrix} \quad (2.130)$$

$$
K_e = \begin{bmatrix}
12\frac{EI_{yye}}{l_e^3} & 0 & 0 & 6\frac{EI_{yye}}{l_e^2} & 0 & 0 & -12\frac{EI_{yye}}{l_e^3} & 0 & 0 & 6\frac{EI_{yye}}{l_e^2} & 0 & 0 \\
0 & 12\frac{EI_{zze}}{l_e^3} & 0 & 0 & 6\frac{EI_{zze}}{l_e^2} & 0 & 0 & -12\frac{EI_{zze}}{l_e^3} & 0 & 0 & 6\frac{EI_{zze}}{l_e^2} & 0 \\
0 & 0 & \frac{EA_e}{l_e} & 0 & 0 & 0 & 0 & 0 & -\frac{EA_e}{l_e} & 0 & 0 & 0 \\
6\frac{EI_{yye}}{l_e^2} & 0 & 0 & 4\frac{EI_{yye}}{l_e} & 0 & 0 & -6\frac{EI_{yye}}{l_e^2} & 0 & 0 & 2\frac{EI_{yye}}{l_e} & 0 & 0 \\
0 & 6\frac{EI_{zze}}{l_e^2} & 0 & 0 & 4\frac{EI_{zze}}{l_e} & 0 & 0 & -6\frac{EI_{zze}}{l_e^2} & 0 & 0 & 2\frac{EI_{zze}}{l_e} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{GJ_e}{l_e} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_e}{l_e} \\
-12\frac{EI_{yye}}{l_e^3} & 0 & 0 & -6\frac{EI_{yye}}{l_e^2} & 0 & 0 & 12\frac{EI_{yye}}{l_e^3} & 0 & 0 & -6\frac{EI_{yye}}{l_e^2} & 0 & 0 \\
0 & -12\frac{EI_{zze}}{l_e^3} & 0 & 0 & -6\frac{EI_{zze}}{l_e^2} & 0 & 0 & 12\frac{EI_{zze}}{l_e^3} & 0 & 0 & -6\frac{EI_{zze}}{l_e^2} & 0 \\
0 & 0 & -\frac{EA_e}{l_e} & 0 & 0 & 0 & 0 & 0 & \frac{EA_e}{l_e} & 0 & 0 & 0 \\
6\frac{EI_{yye}}{l_e^2} & 0 & 0 & 2\frac{EI_{yye}}{l_e} & 0 & 0 & -6\frac{EI_{yye}}{l_e^2} & 0 & 0 & 4\frac{EI_{yye}}{l_e} & 0 & 0 \\
0 & 6\frac{EI_{zze}}{l_e^2} & 0 & 0 & 2\frac{EI_{zze}}{l_e} & 0 & 0 & -6\frac{EI_{zze}}{l_e^2} & 0 & 0 & 4\frac{EI_{zze}}{l_e} & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{GJ_e}{l_e} & 0 & 0 & 0 & 0 & 0 & \frac{GJ_e}{l_e}
\end{bmatrix}
\tag{2.131}
$$

$$
M_e = \left[\begin{array}{cccccc|cccccc}
\frac{156l_em_e}{420} & 0 & 0 & 0 & \frac{22l_e^2m_e}{420} & 0 & 0 & \frac{54l_em_e}{420} & 0 & 0 & \frac{-13l_e^2m_e}{420} & 0 \\
0 & 12\frac{156l_em_e}{420} & 0 & 0 & 0 & \frac{22l_e^2m_e}{420} & 0 & 0 & \frac{54l_em_e}{420} & 0 & 0 & \frac{-13l_e^2m_e}{420} \\
0 & 0 & \frac{2l_em_e}{6} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{l_em_e}{6} & 0 & 0 \\
\frac{22l_e^2m_e}{420} & 0 & 0 & \frac{4l_e^3m_e}{420} & 0 & 0 & 0 & \frac{13l_e^2m_e}{420} & 0 & 0 & \frac{-3l_e^3m_e}{420} & 0 \\
0 & \frac{22l_e^2m_e}{420} & 0 & 0 & \frac{4l_e^3m_e}{420} & 0 & 0 & 0 & \frac{13l_e^2m_e}{420} & 0 & 0 & \frac{-3l_e^3m_e}{420} \\
0 & 0 & 0 & 0 & 0 & \frac{2l_eI_{pe}}{6} & 0 & 0 & 0 & 0 & 0 & \frac{l_eI_{pe}}{6} \\
\hline
\frac{54l_em_e}{420} & 0 & 0 & \frac{13l_e^2m_e}{420} & 0 & 0 & \frac{156l_em_e}{420} & 0 & 0 & \frac{-22l_e^2m_e}{420} & 0 & 0 \\
0 & \frac{54l_em_e}{420} & 0 & 0 & \frac{13l_e^2m_e}{420} & 0 & 0 & \frac{156l_em_e}{420} & 0 & 0 & \frac{-22l_e^2m_e}{420} & 0 \\
0 & 0 & \frac{l_em_e}{6} & 0 & 0 & 0 & 0 & 0 & \frac{2l_em_e}{6} & 0 & 0 & 0 \\
\frac{-13l_e^2m_e}{420} & 0 & 0 & \frac{-3l_e^3m_e}{420} & 0 & 0 & \frac{-22l_e^2m_e}{420} & 0 & 0 & \frac{4l_e^3m_e}{420} & 0 & 0 \\
0 & \frac{-13l_e^2m_e}{420} & 0 & 0 & \frac{-3l_e^3m_e}{420} & 0 & 0 & \frac{-22l_e^2m_e}{420} & 0 & 0 & \frac{4l_e^3m_e}{420} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{l_eI_{pe}}{6} & 0 & 0 & 0 & 0 & 0 & \frac{2l_eI_{pe}}{6}
\end{array}\right]
\tag{2.132}
$$

The general problem allows the computation in a 3D space of structures that may not be aligned with the x-axis as the FEM formulation was developed. So, it is first required to perform a space rotation to ensure that the general reference axes are coherent. This rotation is done using the matrix 2.133 where A represents the matrix with the transformation using the three possible rotation angles: ϕ_{h_e}, ϕ_{v_e} and ϕ_{x_e} for each element as Figure 2.13 shows, and B is a zeros matrix.

$$rot_e = \begin{bmatrix} A & B & B & B \\ B & A & B & B \\ B & B & A & B \\ B & B & B & A \end{bmatrix} \quad (2.133)$$

$$A = \begin{bmatrix} \cos\phi_{h_e} \times \cos\phi_{v_e} & \cos\phi_{x_e} \times \sin\phi_{h_e} & \sin\phi_{h_e} \times \sin\phi_{x_e} \\ & -\cos\phi_{h_e} \times \sin\phi_{v_e} \times \sin\phi_{x_e} & +\cos\phi_{h_e} \times \cos\phi_{x_e} \times \sin\phi_{v_e} \\ -\cos\phi_{v_e} \times \sin\phi_{h_e} & \cos\phi_{h_e} \times \cos\phi_{x_e} & \cos\phi_{h_e} \times \sin\phi_{x_e} \\ & +\sin\phi_{h_e} \times \sin\phi_{v_e} \times \sin\phi_{x_e} & -\cos\phi_{x_e} \times \sin\phi_{h_e} \times \sin\phi_{v_e} \\ -\sin\phi_{v_e} & -\cos\phi_{v_e} \times \sin\phi_{x_e} & \cos\phi_{v_e} \times \cos\phi_{x_e} \end{bmatrix} \quad (2.134)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.135)$$

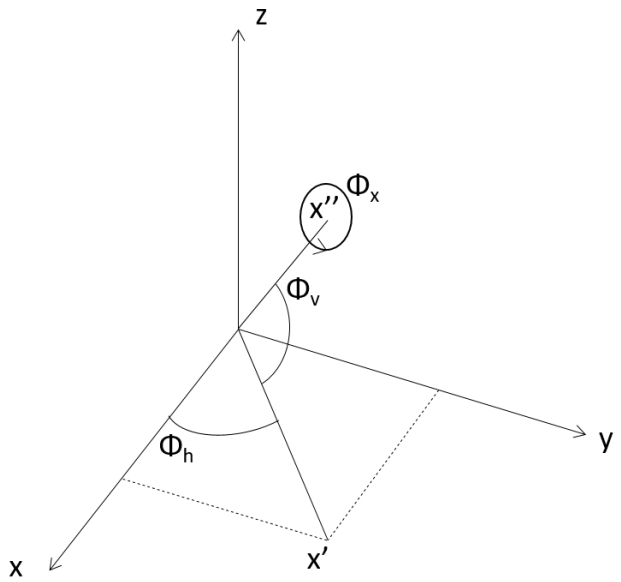

Figure 2.13: Rotation Angles.

This rotation is done by multiplying the stiffness 2.136, mass 2.137 and loads matrices with the

rotation matrix 2.133. The loads matrix does not require this step because the loads are initially inserted assuming the general reference axis.

$$K_e = rot_e^T \times K_e \times rot_e \quad (2.136)$$

$$M_e = rot_e^T \times M_e \times rot_e \quad (2.137)$$

$$F_e = rot_e^T \times F_e \quad (2.138)$$

The last step before the computation of the displacements is to assemble the general matrices for the structure, those matrices contain now all the mesh elements. After assuming an order for the global nodes in the final matrices the principle is to sum, in the correct position of this matrix, the equal degrees of freedom of the nodes of each individual element matrix. If there is not any global node in more than two elements and the nodes are ordered consecutively this step can be done as Figure 2.14 shows, in which each X represents the six DOF of one node.

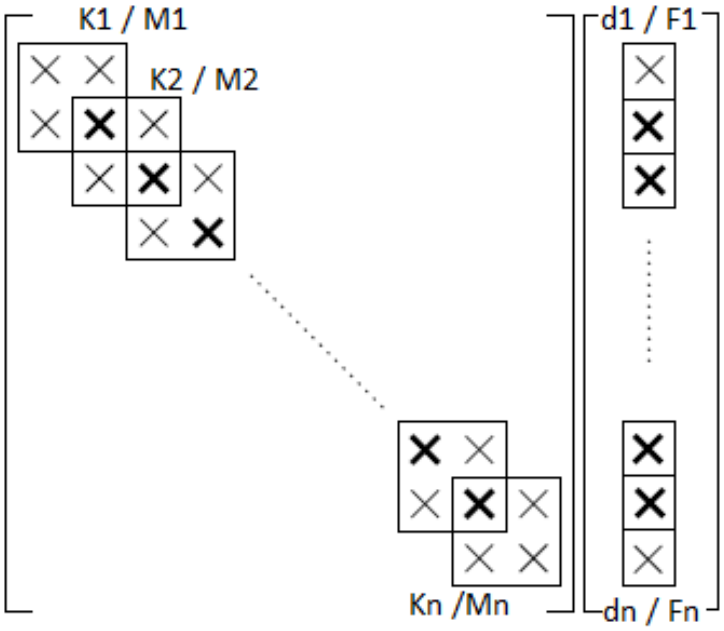

Figure 2.14: Matrices Assembly.

Once having those matrices, the structural displacements problem 2.139 can be solved using methods like: elimination of variables, Cramer's rule or row reduction (also known as Gaussian elimination). In this code the Gauss elimination method is used. The natural frequency free vibrations are calculated using by calculating the eigenvalues ω of the equation 2.140 and the eigenvectors ν. Linear Algebra Package (LAPACK) is a software library that is applied in the code

to solve the two solutions.

$$K \times d = F \quad (2.139)$$

$$K \times \nu = \omega^2 \times M \times \nu \quad (2.140)$$

Chapter 3

Code Development

3.1 Code Structure

This chapter shows how to apply the mathematical model to a computer code. The code should have a simple and clear structure, like Figure 3.1, to avoid errors and allow future developments. So, the current code is divided into the following items:

1. Input data:
    (a) Section geometry;
    (b) Material properties;
    (c) Loads;
    (d) Boundary conditions;
    (e) Structure shape.
2. Section geometry:
    (a) Section properties (section 3.3).
3. Generate mesh;
4. Generate stiffness matrix;
5. Generate mass matrix;
6. Generate rotation matrix;
7. Apply loads in the mesh nodes;
8. Generate loads vector;
9. Apply the problem boundary conditions to the loads and displacements vectors;
10. Perform static analysis to determine the structure displacements;
11. Perform modal analysis to determine the free vibration modes;
12. Show the displaced mesh and structure;
13. Show results.

As it is possible to see in Figure 3.1 the tasks above follow an order that the code must respect to be able to perform a correct analysis.

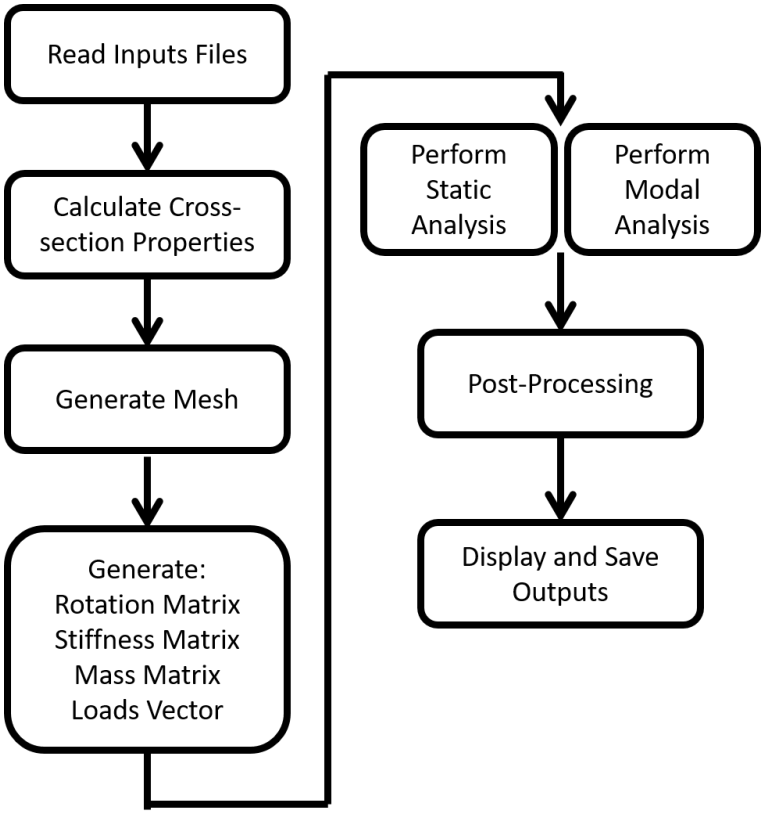

Figure 3.1: Code Flow Chart.

The first part is to load the inputs (see section 3.2) then calculate the cross-section properties and determine the mesh for the analysis. Once having this it is possible to generate the required matrices and vectors that take part of FEM, as stiffness and mass matrices. It is then possible to perform the two analyses and obtain the final results that will be shown in the analysis windows and saved to a file to allow a future comparison between different analysis results. More details about the outputs are explained in section 3.2.

To generate the mesh, the user provides different cross-section shapes and coordinates for its space position and then makes sections with those cross-sections, like in Figure 3.2. Then these sections are divided into small pieces called elements. The number of elements in each section is defined by the fraction of the section length by the total structure length. The nodes coordinates can now be computed assuming that each element has two nodes, one in each extremity, and that the second node of one element is the same of the first node of the following element. It is also assumed that the nodes connect each shear centre of each cross-section. The cross-section properties can now be applied to the correspondent elements.

To generate the stiffness, mass and load matrices the first step is to build matrices 2.131, 2.132 and 2.130 for each element and also the rotation matrix 2.133. These matrices are later reduced to just the current analysis degrees of freedom and the rotation is applied to the element. Then those matrices need to get the full problem size, that is the number of nodes times the number of DOF for the analysis. To achieve that dimensions, the matrices are fulfilled with zeros in the positions of the nodes and DOF that are not taken into account for the respective element. Finally, all the elements expanded matrices are combined into one to form the final matrices.

A practical way to program the above items is to create a main program, like Figure 3.3 shows,

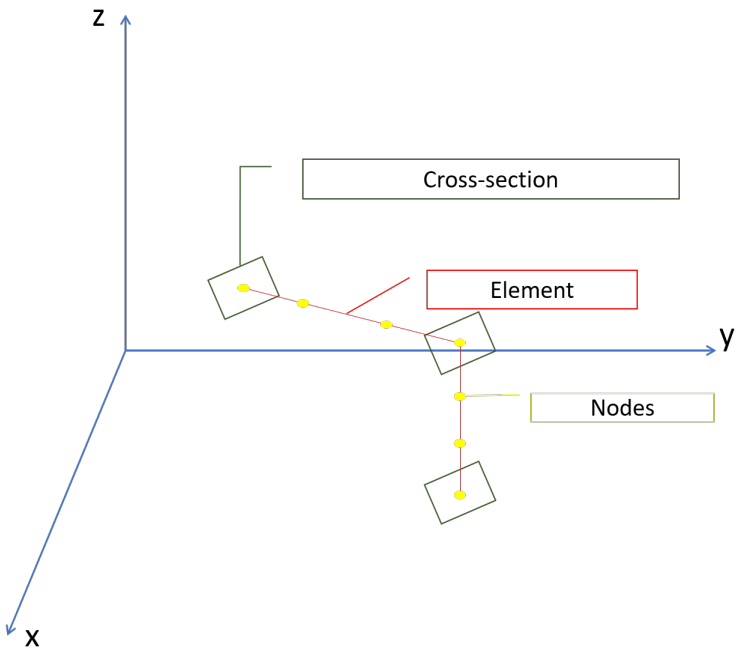

Figure 3.2: Mesh Generation.

from which different subroutine are called. Inside the mesh subroutine other subroutine is called to perform the computation of the cross-section properties that can be different between the elements along the structure. It was also created other Fortran file that stores all the variables. These variables are divided in modules, listed in Figure 3.4, to be easier to identify them.

```
call read_inputs();
call mesh();
call bc_matrix_calc();
call rot_matrix_calculation();
call k_matrix_calc();
call m_matrix_calc();
call f_matrix_calc();
call disp_matrix_calc();
call modal_analysis();
call post_processing();
call outputs();
```

Figure 3.3: Code Main Program.

```
use general_variables;
use degrees_of_freedom;
use cross_section;
use section;
use geometry;
use material;
use nodes;
use elements;
use rotation_matrix;
use boundary;
use loads;
use rigidity_matrix;
use mass_matrix;
use displacement_matrix;
```

Figure 3.4: Code Variables.

3.2 Inputs and Outputs

The inputs for the code are few and quite simple allowing an intuitive understanding of this tool. They are inserted in a proper .txt file called 'inputs' that is read by the code. It is possible to divide the inputs into three different groups: general data (Figure 3.5), structure data (Figure 3.6) and boundary conditions (Figure 3.7) and loads data. A fourth group can be considered, where the cross-section and material properties data bases lists are described, like Figure 3.8 shows, and these are linked with the structure.

```
number of degrees of freedom:
2
degrees of freedom ID (1-w; 2-v; 3-u; 4-teta_y; 5-teta_z; 6-teta_x):
1
4
number of elements:
15
```

Figure 3.5: General data inputs.

The general data indicates the DOF that the user wants to calculate. As already explained, this code allows the computation of six displacements, each displacement is considered as one DOF. Firstly, it is required the number of DOF that will take part in the analysis. Then the user should indicate the numbers of the DOF according to the information provided. It is not necessary to introduce them by ascending order, because the code will organize them that way.

```
number of cross sections
4
cross sections:
ID      x[m]    y[m]    z[m]    geometry[ID]    Material[ID]
1       0.00    0.00    0.00    11              1
2       1.00    0.00    0.00    11              1
3       2.00    0.00    0.00    11              1
4       3.00    0.00    0.00    11              1
number of sections:
3
sections:
ID      point_1[ID]     point_2[ID]
1       1               2
2       2               3
3       3               4
```

Figure 3.6: Structure data inputs.

This is a structural analysis code, so the structure properties are required. First the user must define the number of cross-sections that makes the structure and then provide the coordinates x, y and z of its position in space. It is also fundamental to know the material and the shape of the cross-sections, so the user must provide an identification number (ID) to the desired selections from the corresponding lists bellow, represented in Figure 3.8. Then the number of sections should be provided and after that the ID number of the initial and final cross-sections of each structure section.

Not least important is the loads input data. These inputs are divided in boundary conditions, concentrated loads and distributed loads. The user enters the BC position coordinates and then the conditions are based in a binary code, where one means that the the DOF is free and zero

```
number of boundary conditions (zeros at the correspondent problem's dof):
1
boundary conditions (0-fixed; 1-free):
ID      x[m]    y[m]    z[m]    w       v       u       theta_y theta_z theta_x
1       0.00    0.00    0.00    0       0       0       0       0       0
Number of concentrated loads:
3
ID      x_1[m]          y_1[m]          z_1[m]          x_f[N]          y_f[N]
1       0.0             0.0             0.0             0.0             0.0
2       0.0             0.0             0.0             0.0             0.0
3       3.0             0.0             0.0             0.0             0.0
Number of distributed loads:
2
ID      x_1[m]          y_1[m]          z_1[m]          x_2[m]          y_2[m]
1       0.0             0.0             0.0             1.0             0.0
2       1.0             0.0             0.0             3.0             0.0
```

Figure 3.7: BC and loads inputs.

means it is fixed. To input the concentrated loads it is required to know the desired application position coordinates and the load magnitude in each direction, this load can be a force or a moment. For the distributed loads the principles are the same as the concentrated ones, but it is required to input the initial and final points where the loads are being applied.

```
number of geometries (see image cross-section shapes) (1-airfoil: a-chord[m]; b-inc_ang
14
ID      geometry        a[m]    b[m]    c[m]    d[m]    e[m]    f[m]    file_name
1       1               0.04    0.04    0.00    0.00    0.00    0.00    none
2       2               4.00    4.00    0.10    0.00    0.00    0.00    none
3       3               3.00    3.00    0.00    0.00    0.00    0.00    none
4       4               0.05    0.05    0.005   0.005   0.00    0.00    none
5       5               2.00    3.00    0.00    0.00    0.00    0.00    none
6       6               200     300     10      0.00    0.00    0.00    none
7       7               6.00    8.00    5.00    0.00    0.00    0.00    none
8       8               250     376     150     38      25      38      none
9       9               200     438     250     38      25      100     none
10      10              100     80      20      20      0.00    0.00    none
11      11              0.05    0.05    0.05    0.005   0.005   0.005   none
12      12              200     120     10      10      0.00    0.00    none
13      13              200     120     10      10      0.00    0.00    none
14      0               2000.00 5.00    0.00    0.00    0.00    0.00    aaa.txt
Material properties:
number of materials:
5
ID      material                material density, kg/m3        Young's modulus E, GPa
1       steel                   7850                            200
2       aluminium_alloy         2700                            70
3       pultruded_carbon        1600                            140
4       pine_wood               1600                            140
5       rohacell                1600                            140
```

Figure 3.8: Cross-section and material database inputs.

Once the calculations are done it is necessary to show the results. Some of these results are instantly displayed in the code running windows but these are just the final results from each code section. A .txt file is generated in the code section 'outputs' and it is here that the user can check and confirm all the data generated in the code.

```
--------------------------------
displacement matrix done
--------------------------------
          1   6.28613694911625
          2   39.3945530915280
          3   110.305988483728
          4   216.156343903917
          5   357.324362651096
          6   533.787777664200
```

Figure 3.9: Instant Results Windows.

Other four .DAT files are generated to allow the graphical view of the structure. This files are named as: $'out\_nodes\_initial'$, $'out\_nodes\_final'$, $'out\_section\_initial'$ and $'out\_section\_final'$ and as the names indicate their content is related with the nodes that generate the analysis FEM mesh and the cross-sections coordinates. These files are in Tecplot format so that they can be easily uploaded to this software and the result graphically displayed, as Figure 3.10 shows.

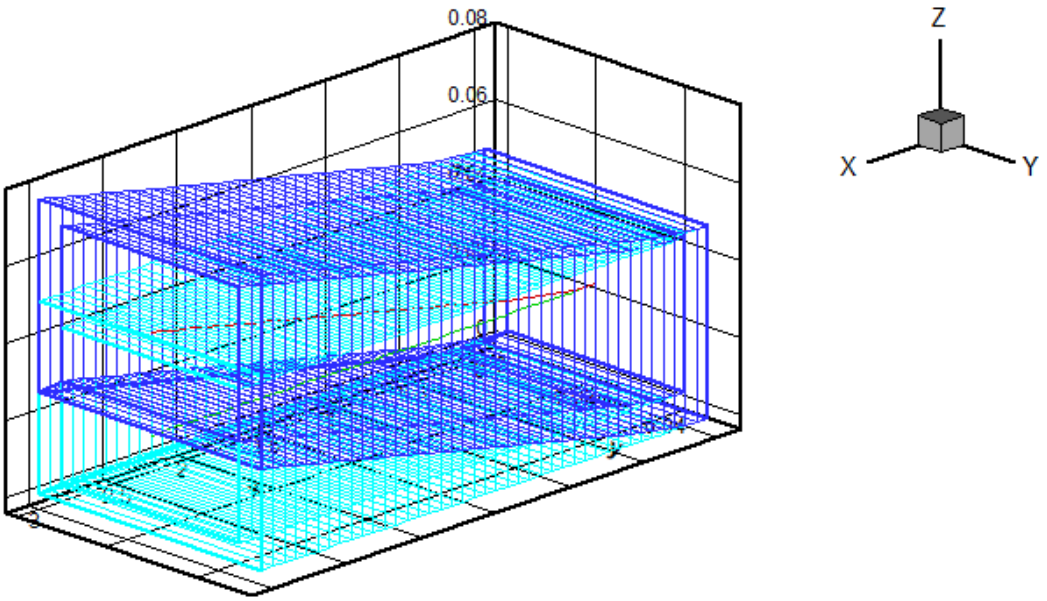

Figure 3.10: Tecplot Results.

3.3 Cross-Section

For this program there have been chosen two different groups of cross section to perform the analysis. The first group is the standard cross-section as: C, T, L, U, I channels and solid and tubular square and circular sections. The second group is the airfoil, that can be solid, as Figure 3.11 shows, or just the skin, like in Figure 3.12. In a future work it will be included airfoil skin section with spars.

Figure 3.11: Solid Section.

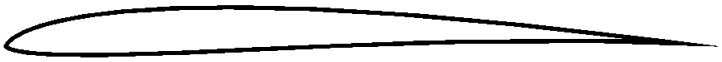

Figure 3.12: Skin Section.

For this cross-sections the inputs, listed in Figure 3.8 are: chord, incidence angle and its position along the chord and the thickness. It is also required a .txt file with the coordinates of the airfoil. If the desired cross-section is one of the standard, the reference letter in the inputs file, on the

cross-section list, in Figure 3.8, matches with a bunch of figures provided, for example in the case of a cross-section C shaped represented in Figure 3.13.

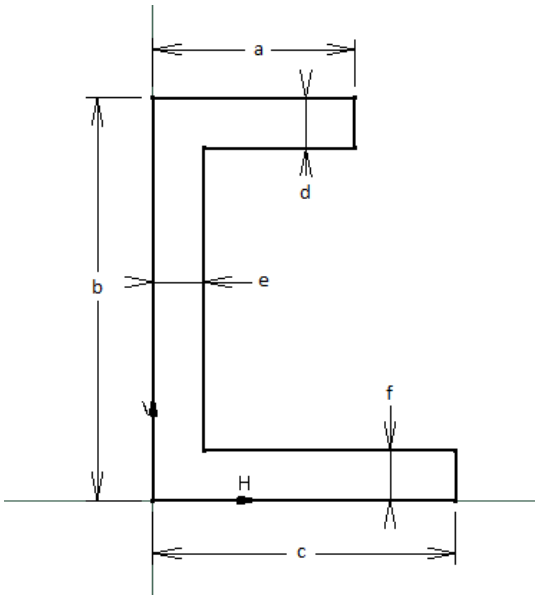

Figure 3.13: C shaped Section.

After studying the mathematical model, it was possible to conclude that to achieve the desired result it is required to have some information about the structure cross-section. This information includes the following items:

1. Area;
2. Geometric centre position (x and y coordinates);
3. Shear centre position (x and y coordinates);
4. First area moments (x axis, y axis and polar or z axis);
5. Torsion constant.

Other inputs to fulfil the FEM initial conditions are the material properties, as already explained in the constitutive model they take an important role in this method. In the cross-section inputs the user must enter one ID for the material, this ID should be coherent with the list of materials, in Figure 3.8. This list has the following items:

1. Material density [ρ];
2. Young's modulus [E];
3. Shear modulus [G];
4. Poisson's ratio [ν];
5. Tensile strength [tu];
6. Compressive strength [cu].

Besides the properties required for FEM the material density is used to calculate the structure mass and the tensile and compressive strengths are not used in this code, although they can be useful for future works, for example, to calculate limit loads or the cross-sections properties in a limit loading case. All these properties should be provided in the pre-established unit of measure and magnitude because the present code does not perform any conversion to these units.

Chapter 4

Code Validation

4.1 Selection of Analysis Cases

After understanding the FEA mathematical model and developing an algorithm to apply the methodology it is necessary to test the code to see if it is reliable and provides valid results.

In order to validate the code, it is required to run some analyses and compare the results to the ones given by some previously validated software. The first step is to choose some cases that are simple to understand and to confirm the results. Then these cases are applied to the developed code and to the Ansys software. After running the solution the results are compared and discussed.

The first three cases are a 3m length steel (see material properties in Table 4.1) beam, with a built-in tip and a 3000N vertical (parallel to z-axis) concentrated load at the free tip. In the code 20 linear elements are selected an therefore the analysis has 21 nodes. In Ansys the mesh generated has 21 quadratic beam elements (BEAM189 Ansys elements) that have 3 node each and that makes a total of 43 nodes. Only the cross-section is different and this difference is enough to change significantly the stiffness matrix values and therefore the results. The principal result in this analysis is the tip displacement and the free vibration modes. Other concern are the cross-section properties values that the code calculates, so this results should also be checked.

4.2 Results Comparison

Firstly, it is necessary to define the material that will be used for the different analysis. It is very important that the material properties are correct because, as it was seen above in the stiffness matrix, they influence directly the final results. So in Table 4.1 it is possible to see the values for different materials. All of them are considered as isotropic because the code only takes into account this type of materials.

Table 4.1: Material Properties.

Property	**Value**			
Name	steel	aluminium	pine wood	Rohacell 110 A
Density ρ [kg/m^3]	7850	2710	500	110
Young's Modulus E [Pa]	200e+009	70e+009	17e+009	0.16e+009
Shear Modulus G [Pa]	76.9e+009	26e+009	5.9e+009	0.05e+009
Poisson's Ratio ν [adm]	0.30	0.34	0.44	0.6

The following tables show a comparison between the code results and an analysis performed with the FEA software Ansys. Several cross-section and structure parameters and the displacement at the structure tip and the six first free vibration frequencies where compared. The difference between the two analysis results is quantified by the deviation (in percentage), where the Ansys results are assumed as the reference values.

4.2.1 Case 1

For the first case the cross-section chosen is a circular shape with 0.04m diameter.

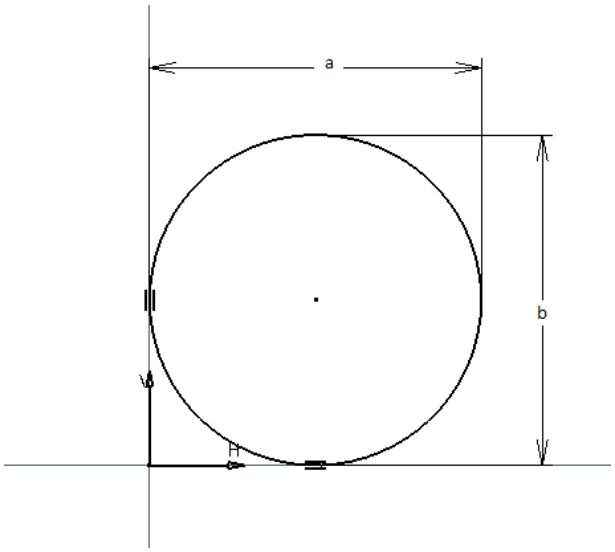

Figure 4.1: Circular cross-section.

Figure 4.1 helps the user to understand the dimensions of the cross-section that will be analysed.

Table 4.2: Case 1 results comparison.

Property	Code Result	Ansys Result	Deviation [%]
Volume [m^3]	3.7699e-3	3.7697e-3	0.0053
Mass [kg]	29.594	29.592	0.0068
Area [m^2]	1.2566e-3	1.2566e-3	0.0
Geometric centre x [m]	0.0	0.0	0.0
Geometric centre y [m]	0.0	0.0	0.0
Second area moment xx [m^4]	1.2566e-7	1.2384e-7	1.469
Second area moment yy [m^4]	1.2566e-7	1.2384e-7	1.469
Second area moment xy [m^4]	0.0	0.0	0.0
Torsion Constant J [m^4]	2.5132e-7	2.4761e-7	1.498
Shear centre x [m]	0.0	0.0	0.0
Shear centre y [m]	0.0	0.0	0.0
Tip displacement [m]	1.0742	1.0766	0.222
Free vibration modes [Hz]	3.138 19.668 55.074 107.938 178.489 266.799	3.1361 19.641 54.943 107.52 177.45 264.6	0.060 0.137 0.238 0.388 0.585 0.831

From Table 4.2 it is possible to conclude that the code is reliable to compute the cross-section

properties of a circular shaped cross-section. The worst result is an error of about 1.5% and it occurs for the torsion constant. Similar error occurs in the computation of the area moments, this happens because Ansys software uses finite elements to approximate the cross-section properties and the code calculates them using analytic equations. The code final results for the tip displacement and for the free vibration frequencies are also very close to the ones from Ansys. It is then plausible to assume that the code is a useful tool for this simple case.

4.2.2 Case 2

In this case a C-channel cross-section is selected.

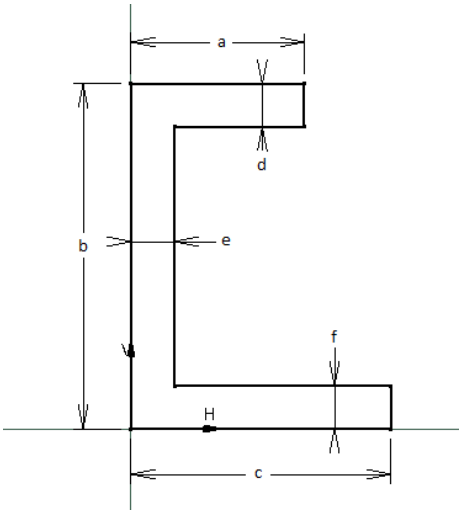

Figure 4.2: C-channel cross-section.

With the help of Figure 4.2 it is possible to fill the input file with the correct dimensions. The sides a, b and c are 0.05m length and the thicknesses d, e and f are equal to 0.005m.

Table 4.3: Case 2 results comparison.

Property	Code Result	Ansys Result	Deviation [%]
Volume [m^3]	2.1e-3	2.1e-3	0.0
Mass [kg]	16.485	16.485	0.0
Area [m^2]	7e-4	7e-4	0.0
Geometric centre x [m]	1.857e-2	1.857e-2	0.0
Geometric centre y [m]	2.5e-2	2.5e-2	0.0
Second area moment xx [m^4]	2.8083e-7	2.8083e-7	0.0
Second area moment yy [m^4]	1.769e-7	1.769e-7	0.0
Second area moment xy [m^4]	0.0	0.0	0.0
Torsion Constant J [m^4]	6.7083e-9	5.8417e-9	14.834
Shear centre x [m]	-1.76711e-2	-1.7475e-2	1.122
Shear centre y [m]	2.53267e-2	2.5e-2	1.306
Tip displacement [m]	0.4807	0.507	5.325
Free vibration modes [Hz]	6.29 39.39 110.31 216.16 357.32 533.79	6.27 39.044 107.96 207.63 334.75 484.39	0.1895 0.8861 2.1767 4.1082 6.7423 10.1983

Once again, looking at Table 4.3, it is viable to use the code to compute the cross-section properties for this case. Almost all the code values have zero difference face to Ansys values, including structure mass and volume. The torsion constant has the largest error, although the order of magnitude is the same, so it is a plausible value for this property. The difference in the torsion constant is explained taking into account that Ansys uses a finite element model of the cross-section to approximate the properties and the code uses a more simple approximation equation that is described in [Bea10]. The tip displacements has a deviation of more than five percent that is a relatively high value.

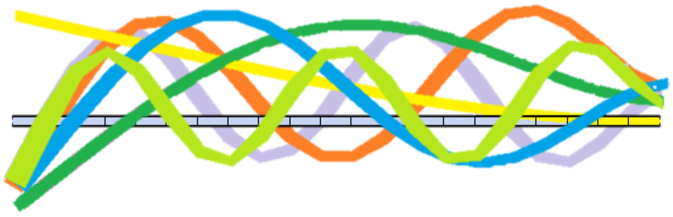

Figure 4.3: Ansys Free Vibration Modes.

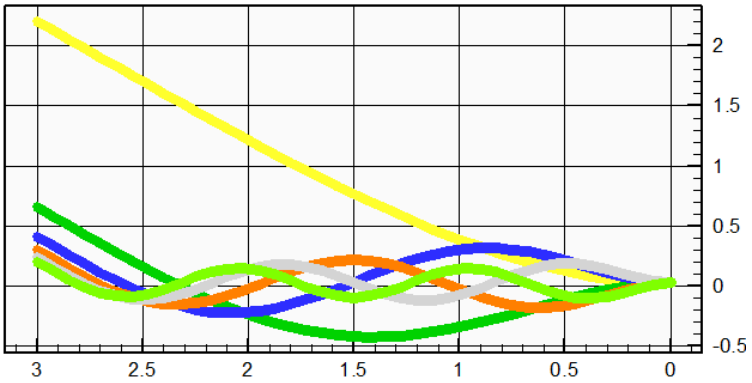

Figure 4.4: Code Free Vibration Modes.

The free vibration modes shapes for the Ansys and code analysis are, respectively, represented in Figures 4.3 and 4.4. In an ascending order of the vibration frequencies, the respective displayed colours, in the Figures, are: yellow, dark green, blue, orange, grey and light green. To guarantee a mesh convergence and better values for the frequencies the number of elements in Ansys was increased to 64 elements (129 nodes). The number of elements in the code analysis was also increased, but the values not change significantly. The free vibration frequencies have an acceptable deviation for the three first modes, although for the higher frequencies modes the deviation between the two values increases to 10% in the sixth mode. These six frequencies are for the vertical displacement DOF, in a future work the respective frequencies for all DOF will also be developed. A plausible explanation for the increasing deviation value lies in the fact that the beam elements used by Ansys ($BEAM189$ element) that have a quadratic interpolation and includes shear deformation effects in the stiffness matrix.

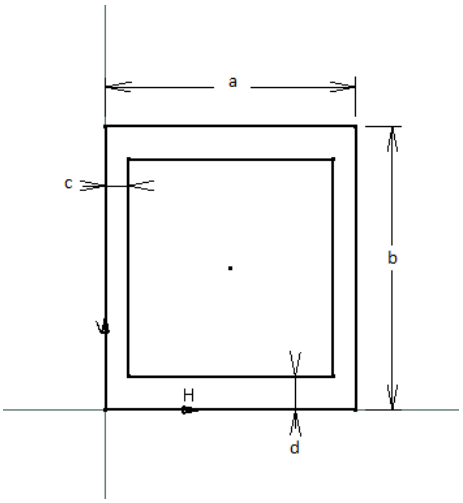

Figure 4.5: Square Tube cross-section.

4.2.3 Case 3

The third case is very similar to the other two. The main difference is the cross-section, that in this case is a square tube with an external side length of 0.05m and 0.005m thickness, as Figure 4.5 shows.

Table 4.4: Case 3 results comparison.

Property	Code Result	Ansys Result	Deviation [%]
Volume [m^3]	2.7e-3	2.7e-3	0.0
Mass [kg]	21.195	21.195	0.0
Area [m^2]	9.0e-4	9.0e-4	0.0
Geometric centre x [m]	2.5e-2	2.5e-2	0.0
Geometric centre y [m]	2.5e-2	2.5e-2	0.0
Second area moment xx [m^4]	3.0750e-7	3.0750e-7	0.0
Second area moment yy [m^4]	3.0750e-7	3.0750e-7	0.0
Second area moment xy [m^4]	0.0	0.0	0.0
Torsion Constant J [m^4]	8.5556e-4	4.8304e-7	1.77e+5
Shear centre x [m]	2.5e-2	2.5e-2	0.0
Shear centre y [m]	2.5e-2	2.5e-2	0.0
Tip displacement [m]	0.4390	0.4393	0.0682
Free vibration modes [Hz]	5.8011 36.3551 101.7998 199.5156 329.9227 493.1569	5.7976 36.205 100.81 195.96 320.7 420.63	0.060 0.414 0.981 1.814 2.875 17.242

The result in this case, displayed in Table 4.4, are very favourable, the tip displacement error is less than 0.1% and the five first vibration frequencies have errors less than 3%. Once again the worst value is related with the torsion constant. The order of magnitude of the torsion constant for this third case is different and that motivates to reconsider a better way to calculate this property. This difference is due to the approximation formula, presented in [Bea10], that the code uses.

4.2.4 Case 4

It is required to perform one more analysis to check if it is possible to compare the code results to the result from an analysis using 3D elements. A structure with 1m length and a circular cross-section with 0.08m diameter is selected for this analysis. The beam material is steel and the boundary conditions are a built-in tip and a free tip and a 2000N vertical distributed force along the beam length. Similar to the previous cases the degree of freedom taken into account is the vertical displacement. In this case 40 elements, therefore 41 nodes, are selected to run the code analysis and the automatic mesh generator in Ansys created 120 SOLID186 (see Figure 5.6) elements that correspond to 725 nodes.

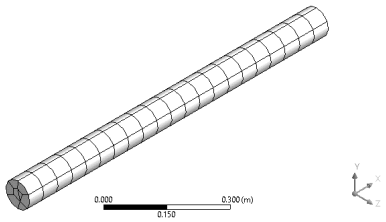

Figure 4.6: Ansys Circular Section Mesh.

In Figure 4.6 it is possible to see the mesh generated by Ansys and check that in this software the elements used are defined in 3D. Looking to the code mesh, in Figure 4.7, it is possible to see that the structure is assumed as a line composed by 1D elements. The green line represents the nodes initial positions and the red line represents the nodes final position after the load has been applied. The code analysis is initially done using only 20 elements, but then this number was increased to 40 to ensure that the results converged.

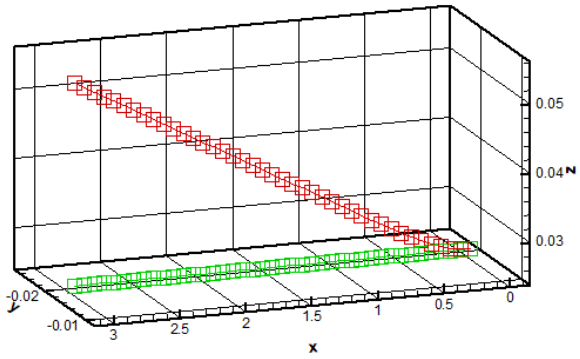

Figure 4.7: Code Circular Section Mesh.

To compare the two analysis, the beam displacements along its length are summarized in a chart displayed on Figure 4.8.

It is possible to see that the two plot lines are almost coincident. In conclusion, there is no significant deviation between a structural analysis computed using 1D elements and a more complex one with 3D elements, for these simple structures. In fact the displacement error at the free beam tip is 0.016%.

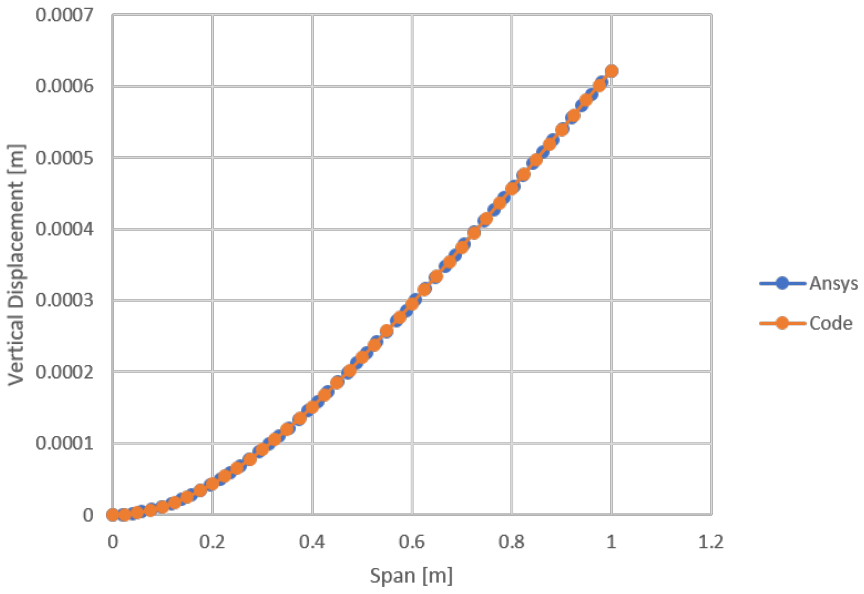

Figure 4.8: Case 4 Results.

Chapter 5

Propeller Structural Analysis

A final code validation is done using the propellers described in [Mor16]. Table 5.1 summarizes the shapes and materials of the different blades that will be analysed.

Table 5.1: Blades Definition.

	Root Airfoil	**Tip Airfoil**	**Root Chord** $[m]$	**Tip Chord** $[m]$	**Radius** $[m]$	**Structure Concept**	**Material**
Blade 1	S1223	S1223	0.2	0.2	1.0	Solid	Aluminium
Blade 2	NACA0012	NACA0012	0.2	0.2	1.0	Solid	Pine Wood
Blade 3	NACA0012	NACA0012	0.2	0.2	1.0	6mm skin	Aluminium
Blade 4	NACA0012	NACA0012	0.2	0.2	1.0	6mm skin Core	Aluminium Rohacell
Blade 5	NACA4412	NACA0012	0.3	0.1	1.0	Solid	Aluminium

Once uploaded the blade airfoils, dimensions, structure concepts and materials to the code input file it is possible to obtain the first results. The cross-section properties are initially calculated and compared with the results determined by [Mor16] from CATIA V5 software. The airfoils used and the respective chords are:

1. S1223 chord 0.2m;
2. NACA0012 chord 0.2m;
3. NACA4412 chord 0.1m.

The results of the cross-section properties are displayed in Table 5.2.

Table 5.2: Cross-section properties comparison using code and CATIA V5.

Blade	**Properties**	**Code Result**	**CATIA Result**	**Deviation [%]**
S1223	I_{xx} $[m^4]$	1.17e-7	1.069e-7	9.448
	I_{yy} $[m^4]$	7.32e-6	5.00e-6	46.4
	gc_x $[m]$	0.06959	0.06942	0.244
	gc_y $[m]$	0.01369	0.01366	0.219
	$Area$ $[m^2]$	0.00259	0.0026	0.384
NACA0012	I_{xx} $[m^4]$	1.087e-7	1.09e-7	0.275
	I_{yy} $[m^4]$	8.035e-6	7.26e-6	10.674
	gc_x $[m]$	0.08425	0.08408	0.202
	gc_y $[m]$	-6.746e-7	0.00000	$\approx$0
	$Area$ $[m^2]$	0.003281	0.003288	0.212
NACA4412	I_{xx} $[m^4]$	7.53e-9	7.55e-9	0.264
	I_{yy} $[m^4]$	5.00e-7	4.54e-7	10.132
	gc_x $[m]$	0.04212	0.04204	0.190
	gc_y $[m]$	0.00309	0.00308	0.324
	$Area$ $[m^2]$	0.000820	0.000822	0.243

Similar to the last chapter the comparison between the two results is done by calculating the error. Looking at Table 5.2 it is possible conclude that the cross-section properties values are in good agreement with CATIA, because the deviations are less than one percent. This had already been concluded in chapter 3.3. Only some second area moments have high errors and this is due to the approximations made by the calculation method of this property.

Other important properties that are fundamental to check are the blade volume and mass. This two parameters are important to define the blades geometries and are summarized in Table 5.3.

Table 5.3: Volume and Mass comparison using code and CATIA V5.

Blade	Code Result	CATIA Result	Deviation [%]
Blade 1 Volume [m^3]	0.002595	0.002597	0.077
Blade 1 Mass [kg]	7.034	7.038	0.056
Blade 2 Volume [m^3]	0.003281	0.003288	0.212
Blade 2 Mass [kg]	1.640	1.644	0.243
Blade 3 Volume [m^3]	0.002444	0.002127	14.903
Blade 3 Mass [kg]	6.6244	5.7632	14.943
Blade 4 Volume [m^3]	0.002444	0.003288	25.669
Blade 4 Mass [kg]	6.6244	5.891	12.449
Blade 5 Volume [m^3]	0.003526	0.003563	1.0384
Blade 5 Mass [kg]	9.5571	9.6556	1.020

Looking at Table 5.3 the errors for the mass and volume of blades 1 and 2 is less than 1% so it is viable to assume that the code can represent structures with a solid airfoil shape cross-section and without taper. Although if the goal is to calculate a skin airfoil shape cross-section, like blade 3, the error is significantly high and this leads to big errors in the analysis displacements. For an airfoil with skin and a core the code does not take into account the core, so the results for the blade 4 are equal to blade 3. This error occured because the code assumes that the airfoil coordinates given by the user are the midpoint of the skin thickness, as in Figure 5.2, when it should be interior to the coordinates, as CATIA V5 does, as represented in Figure 5.1.

Figure 5.1: NACA0012 Skin represented by CATIA V5.

Figure 5.2: NACA0012 Skin represented by Code.

The last blade has an error for both mass and volume of 1% so the code can represent a tapered blade with different airfoils for the root and tip with a good accuracy.

The next step is to analyse the blades with the code and the FEM software Ansys and compare both results. The blades are assumed as built-in and have a free-tip and a 2000N distributed vertical force is applied.

To perform an analysis in Ansys it is first required to draw the different blades in CATIA V5 and

then save them as .igs file format. Then a “Static Structural” Analysis should be selected in Ansys and follow the required steps as Figure 5.3 shows.

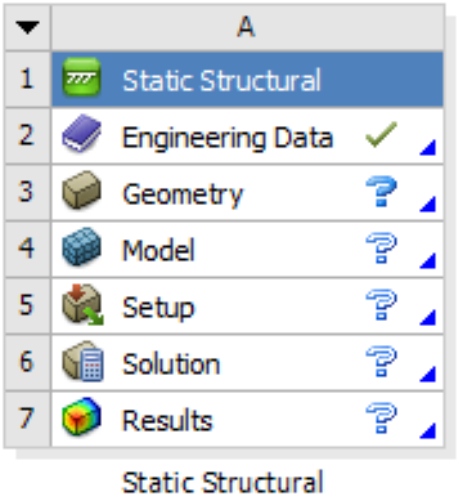

Figure 5.3: Ansys Analysis Steps.

In the “Engineering Data” field, the material properties are loaded. Then in “Geometry” the blades are created, in this case it is just necessary to upload the .igs files that were previously generated. Then in the field “Model” the mesh is generated automatically by the software automatic mesher. The fourth step is to apply the boundary conditions and loads in “Setup”. In Ansys the force is distributed along all the blade surface. Then it is required to choose the desired analysis outputs and this is performed in “Solution”. Here the vertical displacement is chosen. Finally the software can solve the problem and display the results.

```
cross sections:
ID      x[m]    y[m]    z[m]    geometry[ID]    Material[ID]
1       0.00    0.00    0.00    20              2
2       1.00    0.00    0.00    20              2
```

Figure 5.4: Code Analysis Cross-Sections.

ID	geometry	a[m]	b[m]	c[m]	d[m]	e[m]	f[m]	g[adm]	file_name
14	14	0.1	0.0	0.00	0.000	0.00	0.0	0.0	NACA0012.txt
15	14	0.3	0.0	0.00	0.000	0.00	0.0	0.0	NACA4412.txt
16	14	0.2	0.0	0.00	0.000	0.00	0.0	0.0	S1223.txt
17	15	0.2	0.006	0.00	0.000	0.00	0.0	0.0	NACA0012.txt
18	14	0.2	0.0	0.00	0.000	0.00	0.0	0.0	NACA0012.txt

Figure 5.5: Code Analysis Cross-Sections Properties.

To perform an analysis in the developed code, the main step that the user needs to do is to fill the inputs file with the required parameters. To analyse different structures, where the main differences are the cross-section and the material, the user can first input all the desired cross-sections 5.5 and materials and then just select them on the cross-section properties area as seen in Figure 5.4.

The information about the mesh generated by Ansys is summarized in Table 5.4. It is possible to see the number of nodes and elements for each blade. Ansys automatic mesh generation uses two types of elements: a solid type named SOLID186, represented in Figure 5.6, and a surface

Table 5.4: Number of Elements and Nodes for Ansys and code mesh.

Blade	Nodes	Elements SOLID186	Elements SURF54	Elements Code
Blade 1	7125	3534	2039	40
Blade 2	7462	1340	1180	40
Blade 3	2984	480	460	40
Blade 4	4413	680	460	40
Blade 5	6052	1060	1060	40

type named SURF154, represented in Figure 5.7. The solid element is a higher order 3-D 20-node solid element that exhibits quadratic displacement behaviour and has three degrees of freedom per node: translations in the nodal x, y, and z directions [ANS18]. The surface element is used for various load and surface effect applications in 3-D structural analyses and is defined by four to eight nodes and also uses a quadratic interpolation. In the code analysis the user can define the number of linear elements, so each element has two nodes and the number of nodes is equal to the number of elements plus one.

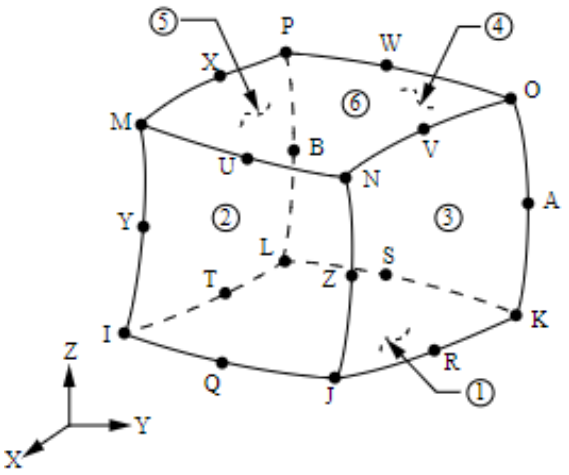

Figure 5.6: SOLID186 Element.

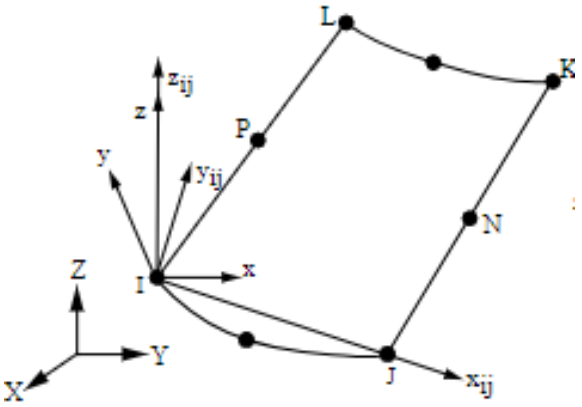

Figure 5.7: SURF154 Element.

After having performed the analysis, it is then possible to compare the results, that are summarized in the following Figures.

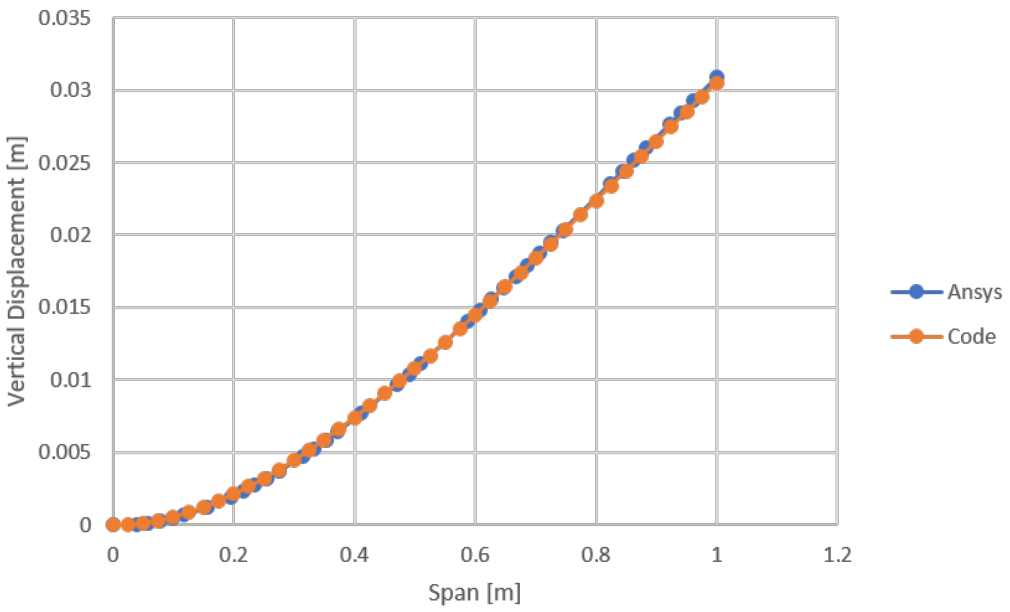

Figure 5.8: Comparison of Bending for Blade 1.

As it can be seen in Figure 5.8, that represents the blade 1 displacement, the two curves are almost identical so, for a structure with two similar solid cross-sections, at each tip, with no tapering and for a distributed load, the code gives valid results. This final result was already expected once the properties also have a low deviation. The deviation between the two displacement at the free tip is 1.27% that is an acceptible value.

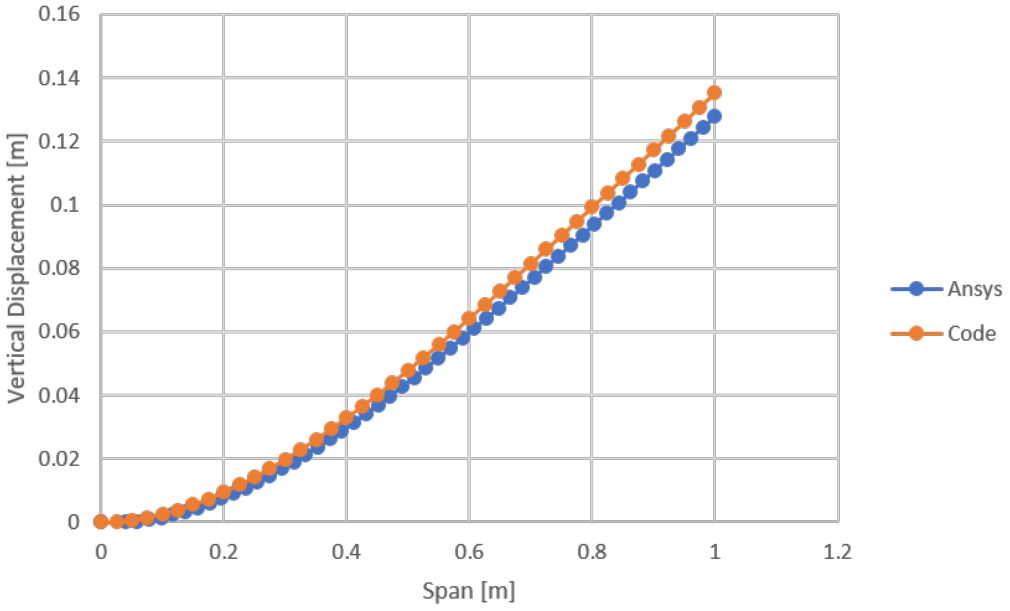

Figure 5.9: Comparison of Bending for Blade 2.

For blade 2 the two analyses results are once again very similar as Figure 5.9 shows. In this case there is no taper and the root and tip airfoils are the same. As it is possible to see in Table 5.3 the blade volume error is higher for this blade than for the previous one, and the same happens with the displacement error. So, once again the difference between the results from Ansys and from

the code can be explained taking into account the initial error in the cross-section properties. Other problem, with the results obtained, is that the values from the code should be lower than the values from Ansys, because this code does not consider the shear deformation that occurs. The displacement's deviation at the tip is 5.81%. This value is in the limit of an acceptable error, so the method should be revised to improve the results from the code's analysis.

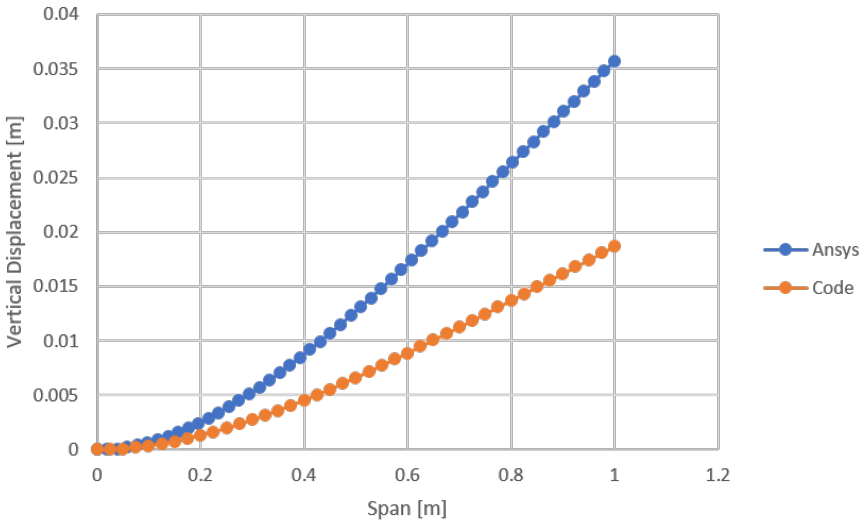

Figure 5.10: Comparison of Bending for Blade 3.

If the user wants to analyse a blade with just a skin the error for the vertical displacement will be very large. At the blade tip the deviation has a maximum value of 47.66% (see Figure 5.10) that is not an acceptable error for a structural analysis. The explanation for this error is, once again, due to the large error obtained previously on the cross-section properties calculation, particularly I_{xx}. In the stiffness matrix these properties have exponents of two and three, so if they are wrong the error will increase significantly.

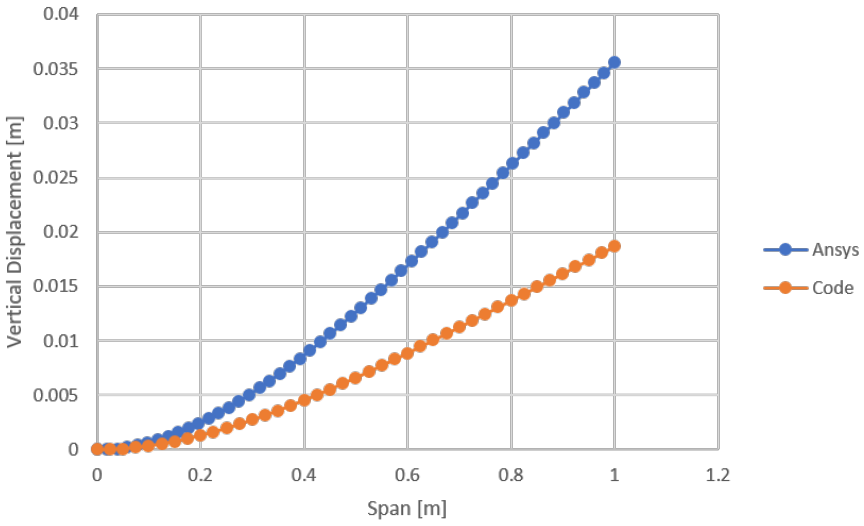

Figure 5.11: Comparison of Bending for Blade 4.

Looking at Figures 5.10 and 5.11 the result are almost the same because the only difference between these two blades are the foam core. This foam core is not taken into account for the

code analysis, so the results are equal for the two blades, and its contribution for the structures stiffness is almost zero so the Ansys results are very similar. The blade displacement deviation for this case is 47.46% and the explanation for this big difference between the two analysis results is the same as for the previous blade.

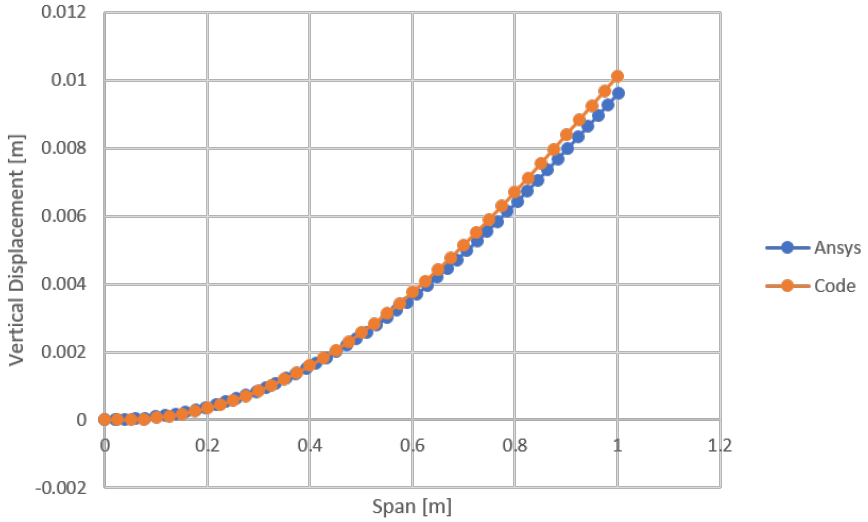

Figure 5.12: Comparison of Bending for Blade 5.

In Ansys the load is being applied over the blade surface, like in the previous cases, that is tapered, so at the blade root the loads are higher than at the blade tip. Contrarily, in the code the load is divided in 10 portions where the load is distributed assuming that the 2000N load is distributed over the surface area and then multiplied by the blade mean chord of each section. This is represented by equation 5.1, where *w is the distributed load applied to the code, F0 is the 2000N load,* c_r, c_t *and* $c(r)$ *are the chords at the blade root, tip and at the required blade radial position,* r, *respectively and* R *is the blade radius.*

$$w(r) = \frac{F}{0.5(c_r + c_t)R} \times c(r) \quad \textbf{(5.1)}$$

As it is possible to see in Figure 5.12, the two curves are almost identical for the first half of the blade radius. For the second half the code results are higher than the Ansys and this can be explained because of an error in the distribution of the load in the code. The code results can be improved if the load is divided into an increased number of portions. At the blade tip the deviation is 5.26%, that is an acceptable value for the structural analysis.

In Figure 5.13 it is possible to see a graphical representation of the initial (green mesh) and final (red mesh) shapes for blade 5. The axis are not in the same scale, otherwise the displacement would not be easy to see.

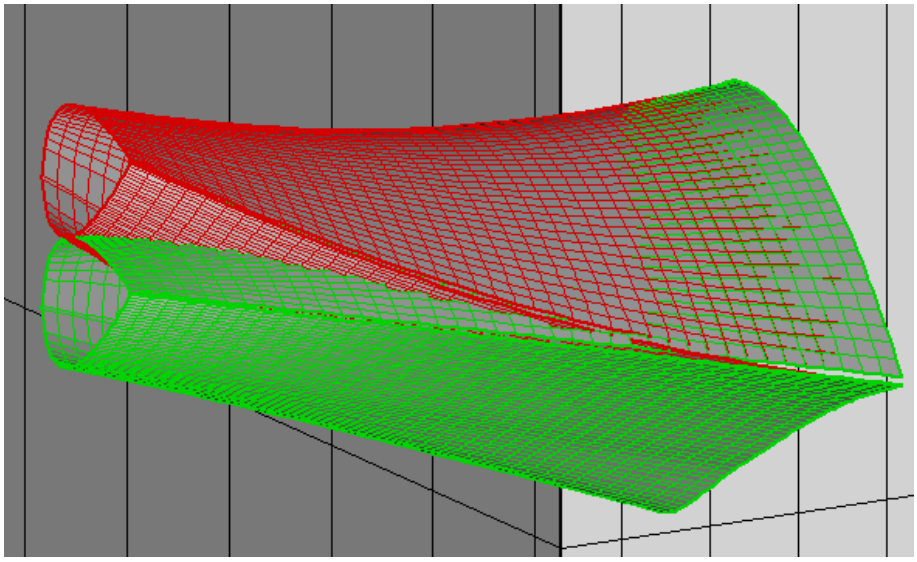

Figure 5.13: Code Graphical Results Display for Blade 5.

Chapter 6

Conclusion

6.1 Conclusions

During this work, a mathematical model for structural 1D finite elements with six, 3D spacial, degrees of freedom was developed. This six DOF are three displacements along and three rotation about each axis x-, y- and z-. This model is divided in three different problems: bar, rod and beam and the combination of them provides the general solution for the structural problem. The result of this task was the stiffness and mass matrices and loads vector and the formulation to calculate the displacements vector and therefore the structure displacements. After this first task, a programming code was created with the mathematical formulation described. This code allows to compute the displacement of a beam structure when subjected to different types of loads and also the free vibration modes. The last step was to check the viability of the code for different structures.

After performing the various analysis the results obtained for the first simple cases are very favourable. The cross-section properties calculated have the expected values, with deviations less than one percent in relation to Ansys software, and therefore the structures displacements and free vibration modes have values that are coherent with the expected values. When the code runs with a cross-section that has an airfoil shape, the results are not good, because there are large deviations in the cross-section properties like the shear centre position, sc_x *and* sc_y*.*

It is possible to conclude that the code works very well for simple cases where the cross-section properties are simple to calculate using equations. For cross-section with complex shapes, like airfoils, the methods to calculate the cross-section properties are not viable and this leads to errors for the final displacements values. A negative point is that the numerical outputs of the displacements require a graphical display. And for that case it is necessary to use a third party software like Tecplot. With this work it was possible to confirm the power of the finite element method to analyse complex structures and see the importance of the stiffness matrix in a structural analysis.

6.2 Future Work

Future works include increasing the number of cross-sections available and allow the study of trusses. Improve the calculation methods for the cross-section properties. Implement a mesh convergence analysis to improve the final results that are dependent of the mesh size. It is also fundamental to add more degrees of freedom to the modal analysis, to have the free vibration frequencies and shapes in 3D space. A graphical users interface to improve the user view of

the inputs and outputs can also be implemented.

Bibliography

[ANS18] *ANSYS. Ansys web-page [online]. 2018. Available from: https://www.ansys.com/ [cited 20 June 2018]. 1, 58*

[Bat14] *Klaus-Jurgen Bather.* Finite Element Procedures, Second Edition. *ISBN: 978-0-9790049-5-7. Prentice Hall, Pearson Education, Inc., 2014. xiii, 5, 7*

[Bea10] *Roy Beardmore. Torsion constant calculation [online]. 2010. Available from: http://www.roymech.co.uk/Useful_Tables/Torsion/Torsion.html [cited 29 October 2010]. 50, 51*

[Das18] *Dassault Systemes. Abaqus web-page [online]. 2018. Available from: https://www.3ds.com/products-services/simulia/products/abaqus/ [cited 20 June 2018]. 1*

[EC13] *A.Pagani E. Carrera. Analysis of reinforced and thin-walled structures by multi-line refined 1d/beam models.* International Journal of Mechanical Sciences, *75:278–287, 2013. 1*

[EC16] *Rehan Jamshed Erasmo Carrera, Alfonso Pagani. Refined beam finite elements for static and dynamic analysis of hull structures.* Computers and Structures, *167:37–49, 2016. 1*

[JS14] *D. Nozak J. Sodja, R. De Breuker. High- and low-fidelity investigations of flexible propeller blades.* AIAA SciTech, *2014. 1*

[Meg07] *T. H. G. Megson.* Aircraft Structures for engineering students. *ISBN-13: 978-0-75066-7395. Elsevier, 2007. 8*

[Mor16] *João Paulo Salgueiro Morgado.* Development of an Open Source Software Tool for Propeller Design in the MAAT Project. *UBI, 2016. 2, 3, 55*

[MSC18] *MSC Software. Patran web-page [online]. 2018. Available from: http://www.mscsoftware.com/product/patran [cited 20 June 2018]. 1*

[OCZ00] *R. L. Taylor O. C. Zienkiewicz.* The Finite Element Method - Volume 1. *ISBN: 0-7506-5049-4. Butterworth Heinemann, 2000. xiii, 5, 6*

[Wik18] *Wikipedia. Fortran wikipedia meaning [online]. 2018. Available from: https://en.wikipedia.org/wiki/Fortran [cited 1 March 2018]. 2*