Study of the vibration behaviour of rotatory blades using the finite element method

Final Degree Thesis

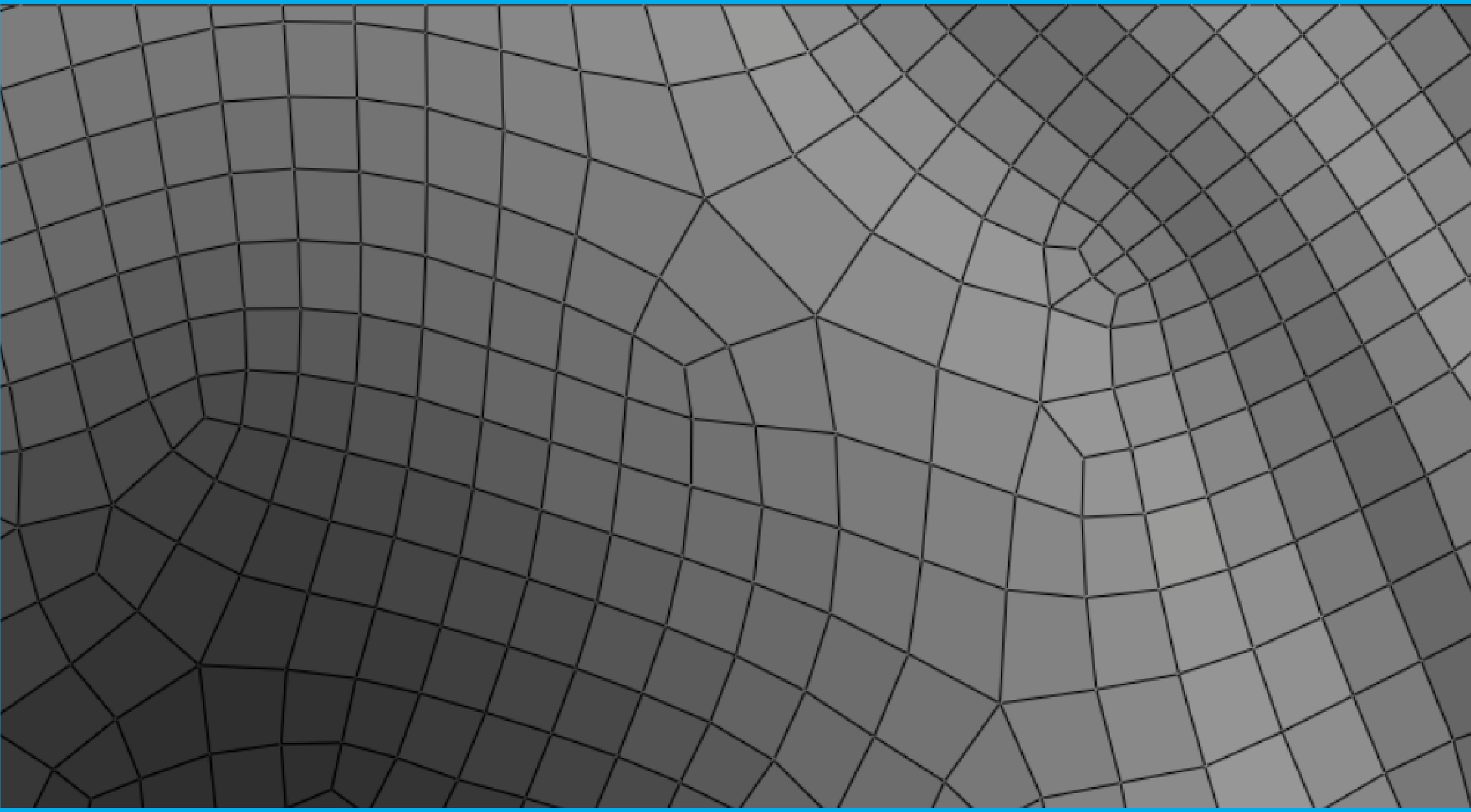
Pere Frontera Pericàs

Bachelor's Degree in Aerospace Technologies Engineering

Director: Hernández Ortega, Joaquín A.

Co-Director: Cante Teran, Juan Carlos

Delivery date: June 10, 2019



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

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

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

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

STUDY OF THE VIBRATION BEHAVIOUR OF ROTATORY BLADES USING THE FINITE ELEMENT METHOD

FINAL DEGREE THESIS

AEROSPACE TECHNOLOGIES ENGINEERING

by

Pere Frontera Pericàs

Director:	Hernández Ortega, Joaquín A.
Co-Director:	Cante Teran, Juan Carlos
Project duration:	February 13 – June 10, 2019

— Report —

Acknowledgments

The present thesis, more than the product of a few months' work, is the culmination of not only four years of Bachelor but also of a long academic trajectory. Getting here would not have been possible without the great deal of support and attendance I've received. For this reason, I would like to dedicate the lines that follow to recognise the work of everyone who, had it not been for them, I could not have reached this milestone.

First, I would like to express my gratitude towards my tutor *prof. Joaquín Hernández* for offering me this unique opportunity. Already standing out as a teacher, as my tutor he has provided me with a renewed sight with regard to investigation, as well as with a handful of tools that I'm sure will be of great utility in my future academic career.

I would like to make a special mention to my high school science teacher for being such a great inspiration. His unparalleled passion and dedication to teach science awakened the curiosity that has ended up taking me here. He has been an exceptional model, in both academic and personal levels. Òscar, if you ever end up reading these lines, *una abraçada molt forta*.

To all my friends and colleagues, thank you for joining me in this journey. Specially to those who have lived along with me these last years, thank you for the lived moments and learned lessons. You have been an unconditional source of motivation.

To my partner, for watching my back and standing me up every time I have fallen, for showing me the value of effort and perseverance, thank you.

A very special thanks to my family, for bringing me up in such a cosy household. To my cousins, for having welcomed me since I was a kid. To my mother, for all the human values she has taught me. To my father, for being my reference engineer. To my brother, for having so much confidence deposited on me. Growing up with you has been the adventure of a lifetime.

And finally, a deep acknowledgement to all the people who has contributed in whatever way to take me where I am now. As a wise scientist once said:

"If I have seen further it is by
standing on the shoulders of Giants"

Abstract

The physics lying behind rotordynamics is complex to model, so that in many cases numerical processing is the only feasible approach. Being rotordynamics a field of great interest in the aerospace industry, the efforts devoted to its understanding are increasing day by day. Following this tendency, the aim of the present study is to develop a simplified elastodynamic model for the case of rotating structures such that can be addressed through numerical tools, built using the finite element method. For the purpose of analysing the vibration phenomena, modal decomposition and numerical integration have been taken advantage from. In this context, it has been found that the singular value decomposition could be applied in structural analysis to extract dominant displacement fluctuations, allowing the unfolding of global properties of the dynamic response. In the present report, the singular value decomposition has been applied to cantilever beams undergoing a single rotation, giving rise to reasonably satisfactory results.

Keywords : *blades, FEM, rotordynamics, SVD, vibration*

Contents

List of Figures

List of Tables

List of Symbols

The next list describes several symbols that will be later used within the body of the document.

Abbreviations

CAD	Computer Aided Design
DOF	Degree Of Freedom
FEM	Finite Element Method
SVD	Singular Value Decomposition
RSV	Right Side Vector
LSV	Left Side Vector

Mathematical notation

i	Imaginary unit
$\mathbf{I}$	Identity matrix
$\mathfrak{L}$	Lagrangian
n	Outward normal unit vector
$\mathbb{R}$	Set of real numbers
$\mho$	Domain
Γ	Boundary
$\varnothing$	Null space
∇^s	Symmetric gradient
$\tilde{\square}$	Spin of a vector
$\square^T$	Transpose
$\dot{\square}$	First temporal derivative
$\ddot{\square}$	Second temporal derivative
$\square^e$	Elemental variable
$\square_s$	Variable expressed in the s frame

General variables

A	Area
ρ	Density
F	Force
$\mathbf{J}$	Inertia Tensor
m	Mass
M	Moment
P	Power
R	Radius
t	Time
T	Torque

Coordinate systems

$\boldsymbol{x} = \left\{x\, y\, z\right\}^T$	Inertial frame
$\boldsymbol{\varsigma} = \left\{\varsigma\, \varrho\, \varpi\right\}^T$	Corotational frame
$\boldsymbol{\xi} = \left\{\xi\, \eta\, \zeta\right\}^T$	Parent domain

Kinematics

a	Acceleration
α	Angular acceleration
θ	Angular position
Ω	Angular velocity
x	Position
$\mathbf{Q}$	Rotation matrix
$\mathbf{T}$	Rotation tensor
$\mathbf{U}$	Total displacements
v	Velocity

Elasticity

k	1D stiffness
$\mathbf{u}$	Elastic displacements in $\boldsymbol{\varsigma}$
$\mathbf{R}$	Elastic reactions
$\mathbf{C}$	Elasticity tensor
$\mathfrak{u}$	Generic elastic displacement
ν	Poisson's coefficient
$\bar{\mathbf{t}}$	Prescribed tractions
$\overline{\alpha}, \overline{\beta}$	Rayleigh damping parameters
$\mathbf{W}$	Reaction torque
I	Second moment of area
$\boldsymbol{\varepsilon}$	Strain tensor
$\boldsymbol{\sigma}$	Stress tensor
E	Young's modulus
$\overline{\mathbf{D}}$	Viscoelasticity tensor

Finite element method

$\mathbf{A}$	Assembly operator
$\mathbf{H}$	Circulatory matrix
$\mathbf{CN}$	Connectivity matrix
$\mathbf{D}$	Damping matrix
$\mathbf{J}^e$	Elemental Jacobian
$\mathbf{f}$	Force vector
ξ_g	Gauss point
w_g	Gauss weight
$\mathbf{d}$	Global displacements vector
$\mathbf{c}$	Global variations vector
$\mathbf{G}$	Gyroscopic matrix
$\mathbf{M}$	Mass matrix
$\mathbf{N}$	Matrix of shape functions
$\mathbf{B}$	Matrix of the gradient of $\mathbf{N}$
n^e	Nodes of the element e
n_{el}	Number of elements
n_{pt}	Number of nodes
n_d	Problem dimension
$\mathcal{R}$	Rigid body modes
$\mathbf{l}$	Set of unconstrained DOFs
$\mathbf{r}$	Set of constrained DOFs
$\mathbf{K}$	Stiffness matrix
$\mathcal{V}, v_i$	Test functions
$\mathcal{S}, u_i$	Trial functions

Modal analysis

$\bar{\xi}_i$	Damping ratio
λ_j	Eigenvalue
$\mathbf{q}$	Generalised coordinates
ω_j	Natural frequency
$\boldsymbol{\Phi}_j$	Natural mode

Numeric integration and SVD

β	Explicit character parameter
$\mathbf{U}$	Matrix of left side vectors
$\mathbf{V}$	Matrix of right side vectors
$\boldsymbol{\Sigma}$	Matrix of singular values
γ	Numerical viscosity
$\Delta \mathrm{t}$	Timestep

Aim

The aim of this project is to develop a MATLAB software capable of solving the differential equations describing the dynamic behaviour of rotatory blades using the finite element method. Once the software is implemented, it will be combined with the singular value decomposition to study rotating structures in nonlinear regime, emphasising the determination of predominant modes and frequencies.

This software would serve as a generalisation of the finite element theory learnt during the Bachelor's degree for rotatory and non-inertial frames of reference, allowing the implementation of the method to some fields of aeronautical interest.

Scope

In order to fulfil the aim of the project and ease its development, the work has been split into the following tasks:

- Getting familiar with the mechanics associated to non-inertial reference systems and its formulation in terms of the Finite Element Method.
- Posing the differential equation that governs the behaviour of the system and formulating its weak and strong form.
- Development of a software capable of solving the latter equation using typical FEM discretisation.
- Comparison between numeric integration and modal analysis.
- Implementation of the singular value decomposition and application to rotating beams.
- Results visualization, using postprocessing software or other animating methods.
- Modelling of the main aerodynamic forces acting on the blade. Further analysis and result comparison.
- Convergence test to ensure software verification.

Those tasks are meant to produce a document with the following structure:

- **Report**: introduction, development and results.
- **Report attachments**: particular topics, raw simulation results and MATLAB code.
- **Budget**.

Requirements

The final documents and numerical tools are to meet the following specifications:

- The developed software will be coded in MATLAB.
- The study will be restricted to linear elasticity and small strains theory.
- The code must be able to display the temporal evolution of the system.
- For the 3D case, the blade geometry has to be introduced from a CAD software.
- The program must run in a reasonable amount of time with a standard computer (i7 -5500U 2.4 GHz – 16 GB RAM).

1 Introduction

The aim of this section is to briefly introduce the reader to the field of **rotordynamics** and outline the main models used nowadays to tackle these kind of problems. An overview of the report structuring will also be presented, as well as a review on the actual possibilities to develop the present work. This introduction follows the key concepts presented in [1], [2] and [3] .

1.1. Historical frame and background

Since the invention of the potter's wheel, almost 6000 years from now, humanity has relied on rotating machinery for a wide range of activities and processes. This idea of a symmetrical object turning on an axis is the key behind complex inventions such as mills, water wheels and even engines and jet turbines.

Behind those great inventions lies a great understanding of the physics describing this rotational motion. **Greeks** were the first who tackled machinery in a systematic and even mathematical point of view, bringing new inventions such as Archimedes' screw and Hero's *aelopile*, the first reaction turbine — which unfortunately could not produce useful work.

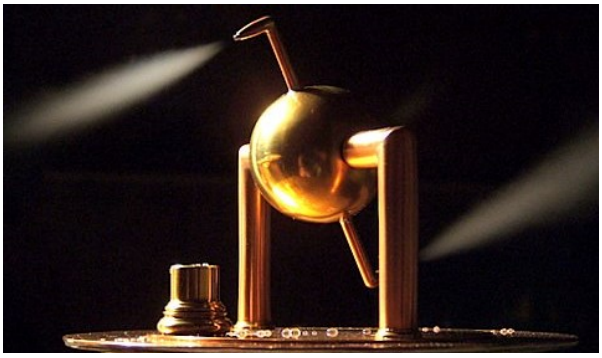


Figure 1.1: Replica of Hero's *aelopile* [4]

Water wheels became an important source of energy in ancient Chinese civilisation, and later on in the Roman empire as a substitute for manual and animal work. It was used for a wide range of applications, including iron casting and grain grinding. Wind mills took longer to appear, the first practical wind-powered machines dating from the 9^{th} century in Persia, and reaching Europe three centuries later. These kind of rotating machinery were, however, quite basic, and needed from better scientific background in order to be meaningfully developed.

The world had to wait until the Renaissance Period to see substantial contributions to this field. Leonardo da Vinci is credited for some fundamental contributions to solid mechanics, which were improved by Galileo and later on by Euler and Bernoulli, who formulated the correct equations for simple bending. In the zenith of **scientific revolution**, Newton's development of calculus unfold a new range of applications of mathematical methods to engineering. It was thanks to the latter that Daniel Bernoulli could formulate the differential equation of motion for a vibrating beam, to be later accepted by Euler when investigating beams under different loading conditions.

The Euler-Bernoulli beam theory was improved by Rayleigh (1877) and later on by Timoshenko (1921), adding the effect of shear. This beam theory, valid even today in the 21^{st} century, is the backbone of all rotor analysis. The two-dimensional study of structures was carried by Germain and Kirchhoff (1850), and lucid discussions about vibrations of elastic systems were carried by Lord Rayleigh.

Thanks to the contribution of these and many other scientist and mathematicians,the fundamentals of the **Theory of Elasticity** could be established during the scientific revolution. These deformable bodies were referred as structures by engineers. As the industrial revolution began and the first steam and internal combustion engines appeared, engineers needed from approximate methods, based on energy principles, to carry out accurate studies for solid structures with intricate geometry.

By then, as the required speed of the machinery was rather slow, reciprocating machinery could be used instead of rotating devices. In fact, little was known about the mechanical behaviour of beams under rotating forces. A first study on the topic was carried by Rankine (1869), proposing that a critical speed exists above which rotation becomes impossible. This was later denied by **De Laval** (1883), a pioneer in the design of steam turbines.

Once the physics of rotordynamics were well understood and steam could be used as motive force, there was a great improvement in the capacity of power generation. The first reaction turbine came up in 1884 at the hands of Charles Parsons. Later on, great development was achieved in the field of rotordynamics due to the invention of the Dynamo in 1878, which led to an outstanding expansion of the steam turbine.

Little time had to pass for rotating machinery to be used as source of **propulsion**. The first turbojet-powered aircraft was the *Heinkel HE-178*, dating from 1939. Since then, high-speed complex rotating machinery has taken an important role in many industrial and aeronautical fields. However, they require from precise estimation to work properly, as they are submitted to high thermal and stress loads. Elasticity theories and energy methods are relatively easy to apply to bars, beams, discs, plates and even membranes, but when it comes of complex three-dimensional structures, no analytical solution exists.

In this context, **numerical analysis** arose. Until the half of the past century, analysis of complex structures was already been carried in a numerical approach using the matrix method. However, even tough it gave good results for simple structures under static conditions, the emergence of aeronautics uncovered some of its main drawbacks, such as the impossibility to describe the flutter phenomena.

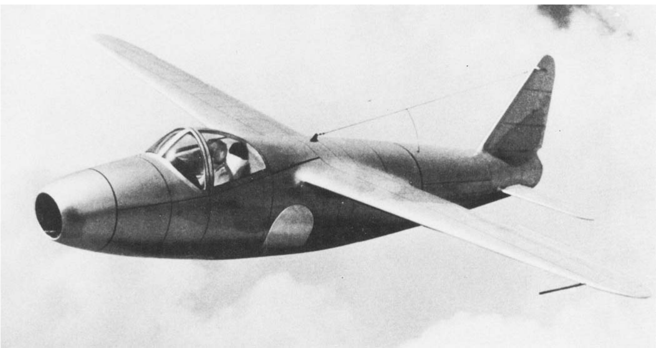


Figure 1.2: Picture of the *Heinkel HE-178* [5]

With the release of NASTRAN by the end of the sixties and further publications pointing its mathematical basis, the **finite element method** gained popularity among physicists and engineers for its wide range of application. Nowadays, its main drawback is the computational cost, as the computation time increases with the number of elements to simulate.

This problem has already been addressed by applying modal decomposition analysis, which provides great results with less computation cost for dynamics systems under harmonic excitations. Unfortunately, physics describing the vibration phenomena in rotating structures are highly nonlinear, and thus alternative paths are to be sought. The **singular value decomposition** seems to be a good candidate that, in combination with modal analysis and numerical integration, could unfold predominant patters of the structure. If applied to rotordynamics, some phenomena of aeronautical interest could be studied, including the vibrations generated in helicopters' and wind turbines' blades. Other effects caused by the rotation conditions could also be addressed, such as the centrifugal stiffening and resonance phenomena. The latter is, in fact, one of the fields of investigation that lead to the foundation of the finite element method.

1.2. State of the art

This section reviews some of the basic topics of rotordynamics and introduces the reader to some essential but important concepts and models related with the dynamics of rotating structures.

1.2.1. Introduction to rotordynamics

Rotordynamics is the branch of applied mechanics dealing with devices in which at least one part, the *rotor*, rotates with significant angular momentum. Although the definition of rotor states that a set of hinges or bearings constrain the position of the rotation axis, there are some cases in which the rotor is not restricted at all. These are known as *free rotors*, with a spin speed governed by conservation of angular momentum. Examples of free rotors are spinning projectiles, space vehicles and even celestial bodies such as spinning neutron stars.

The aim of this report is, however, to study the so-called *fixed rotors*, in which the spin axis is somehow fixed and the spin speed imposed by a **driving device**. First rotordynamic studies date from the late 19^{th} century, when many methods to deal with rotatory machinery were developed (see A.2). One of the simplest models that described the behaviour of rotatory structures was the **Jeffcott rotor**, which can qualitatively explain many important features of real-life rotors. However, this simple model fails in computing precise values and in modelling complex machinery, as cannot predict some phenomena such as the dependence of the natural frequencies on the rotational speed.

As time passed by, higher power density, lower weight and faster machinery were demanded. As a consequence, correct quantitative prediction became particularly important. Torque is usually the critical factor when sizing the machine, so in applications related with power generation higher rotation speed is a must. With the increase in speed, power devices became lighter by using stronger but lighter materials. However, stronger doesn't always mean stiffer, an these lighter structures were more prone to **vibrate**.

Another field that demanded high performing features was propulsion, which needed from high thermodynamic efficiency and power density. That meant that the same amount of heat had to be generated in a smaller volume, and hence with lower thermal capacity. Rotating machines working under these conditions not only have to handle high temperatures, but also stresses due to the thermal gradient.

Although there has been an outstanding progress in the field of rotordynamics during the recent years, it is still a field of very **active research**. Attitude control of space vehicles, design of wind turbines, industrial rotating machinery, turbojets, electric motors, power generators, helicopter's rotors and space launchers (rockets) are only a few of all the engineering fields that could benefit from those investigations in the near future.

1.2.2. Linear rotordynamics

Rigid body equations

Imagine a rigid body rotating and moving around, with mass m and principal moments of inertia J_ς, J_ϱ, and J_ϖ referred to a frame $\varsigma\varrho\varpi$ fixed to the body and coincident with its principal axes of inertia. The equations describing its motion are quite complex, with non-linearities in the rotational degrees of freedom. Being xyz the inertial frame of reference, the equations of motion under the effect of a moment **M** and force **F** are:

$$\mathbf{F} = \mathrm{m}\frac{d\,\mathbf{v}}{d\mathrm{t}} = m\,\mathbf{a} \ \ , \ \ \mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times (\mathbf{J}\boldsymbol{\Omega}) = \mathbf{M} \tag{1.1}$$

Which correspond to Newton's second law and Euler's equation for rigid body dynamics, being $\boldsymbol{\Omega}$ the angular velocity of the body and **a** its linear acceleration. Splitting into components, yields to:

$$\begin{cases} m\ddot{x} = F_x \\ m\ddot{y} = F_y \\ m\ddot{z} = F_z \end{cases} \qquad \begin{cases} M_\varsigma = \dot{\Omega}_\varsigma J_\varsigma + \Omega_\varrho \Omega_\varpi \left(J_\varpi - J_\varrho\right) \\ M_\varrho = \dot{\Omega}_\varrho J_\varrho + \Omega_\varsigma \Omega_\varpi \left(J_\varsigma - J_\varpi\right) \\ M_\varpi = \dot{\Omega}_\varpi J_\varpi + \Omega_\varsigma \Omega_\varrho \left(J_\varrho - J_\varsigma\right) \end{cases} \tag{1.2}$$

As can be spotted, Euler equations are **nonlinear** in the angular velocity. Moreover, the previous equations only describe the behaviour of a rigid body and elasticity plays no role whatsoever. Thus, one can state that the equations describing the behaviour of an structure under rotational conditions will be highly nonlinear.

However, a linearised model describing the basic features of rotating systems can be obtained embracing some **simplifications** :

- The rotor has a rotation axis that coincides with one of its baricentrical principal axes of inertia (the rotor is perfectly balanced).
- The deviations from the previous condition are assumed to be small.

- Displacements and velocities (linear and angular) are assumed to be small, with the exception of the rotation angle and angular velocity about the spin axis, which in many cases will be imposed by the driving system.

Equation of motion

If the previous simplifications are taken into account, a rotor with **constant** spin velocity Ω can be modelled discreetly by using the adequate elastic model, and the following dynamic equilibrium equation is obtained:

$$\mathbf{M}\,\ddot{\mathbf{q}}(\mathbf{t}) + (\mathbf{D}+\mathbf{G})\,\dot{\mathbf{q}}(t) + (\mathbf{K}+\mathbf{H})\,\mathbf{q}(t) = \mathbf{f}(t) \tag{1.3}$$

In the previous equation, some of the present terms may look familiar to the reader: **M**, **K** and **D** stand for the symmetric mass, stiffness and damping matrices, while $\mathbf{q}(\mathbf{t})$ is a vector containing the generalised coordinates referred to the inertial frame xyz and $\mathbf{f}(\mathbf{t})$ a time-dependant vector that accounts for the effect of external forces. However, in the context of rotordynamics, new matrices appear: **G** and **H** are the skew-symmetric gyroscopic and circulatory matrices, respectively.

The **gyroscopic** matrix contains inertial and conservative terms, linked with the gyroscopic moments that act on the rotating elements of the structure. If the equations are referred to the rotating and thus noninertial frame of reference, the **Coriolis** effect is included in the gyroscopic matrix. The **circulatory** matrix, on the other hand, contains the nonconservative effects related with internal damping of the rotating parts and, in some cases, accounts for the effect of the surrounding fluids. In many cases, it is found that in the field of rotordynamics, both **G** and **H** are proportional to the spin velocity Ω, and when the latter tends to zero, the gyroscopic and circulatory skew-symmetric elements vanish and the system reduces to that of a static structure.

It is important to notice that in this context, **K** and **D** may not have the same expression they had in the case of still strictures. In fact, they can be split into two matrices: one corresponding to static structure and another which depends on the rotating speed, often on Ω^2.

Complex coordinates

In some cases, the simplifications made in the previous page cannot be assumed, and the study of the rotor becomes very complicated. In this situations, however, an model resembling equation 1.3 can be obtained, although with reference to a **noninertial frame**. Moreover, most flexible rotors can be assumed to behave as beams. With these hypotheses, the lateral behaviour of the rotor can be uncoupled from its axial and torsional, simplifying thereby the resulting equations. Mathematically, this is done by using complex coordinates.

If the problem can be reduced into two-dimensional, that is, spinning around the z axis and displacements only in the xy plane, the latter can be expressed in terms of a complex number $r(t) = x(t) + iy(t)$, where i stands for the imaginary unit $\sqrt{-1}$. This formulation is equivalent to representing displacements in vectorial form, but allows a more convenient analytical form to solve the rotor equation of motion (eq. 1.3).

The key of this method is that symmetric matrices are real when the equations are written in terms of complex coordinates, whereas skew-symmetric matrices become symmetric but with imaginary terms. Using the complex number formulation, the equation of motion for a axisymmetrical rotor results in:

$$\mathbf{M}'\,\ddot{\mathbf{q}}'(t) + \left(\mathbf{D}' + i\mathbf{G}'\right)\dot{\mathbf{q}}'(t) + \left(\mathbf{K}' + i\mathbf{H}'\right)\mathbf{q}'(t) = \mathbf{f}'(t) \tag{1.4}$$

Separating real from imaginary parts:

$$\begin{bmatrix} \mathbf{M}' & \mathbf{0} \\ \mathbf{0} & \mathbf{M}' \end{bmatrix} \ddot{\mathbf{q}}(t) + \left(\begin{bmatrix} \mathbf{D}' & \mathbf{0} \\ \mathbf{0} & \mathbf{D}' \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\mathbf{G}' \\ \mathbf{G}' & \mathbf{0} \end{bmatrix} \right) \dot{\mathbf{q}}(t) + \left(\begin{bmatrix} \mathbf{K}' & \mathbf{0} \\ \mathbf{0} & \mathbf{K}' \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\mathbf{H}' \\ \mathbf{H}' & \mathbf{0} \end{bmatrix} \right) \mathbf{q}(t) = \mathbf{f}(t) \quad (1.5)$$

The solution of the previous system can be written in terms of the complex frequency $\mathbf{s} = \sigma + \mathbf{i}\omega$: the imaginary part ω stands for the frequency of the free motion (*whirl* frequency) while the real part σ is the decay rate changed in sign.

The main advantage of this method is that allows the system to be analysed using **modal decomposition**, which in conventional coordinates required from all the matrices to be square and symmetrical. Further details on the modal analysis and computation of eigenvectors and eigenvalues can be found in section 5.1.

The Campbell diagram

As has been seen, the spin speed of the structure can appear explicitly in the equation of motion. Thus, a dependence of the natural frequencies on this speed is expected. If that is the case, it is quite common to **plot** the natural frequencies ω as functions of Ω. In many cases, frequencies characterising exciting forces also depend on Ω and so are reported in the same graph, which is commonly known as *Campbell diagram*. Clearly, it is symmetrical for both ω and Ω axis, so only the first quadrant is represented. The intersections with the ω axis correspond to the natural frequencies at standstill of the system ($\Omega = 0$). It is important to note that the Campbell diagram can only be graphed if the equation of motion is linearised, because only in this case the concept of natural frequencies applies.

If the exciting forces are time-dependent with an harmonic time story, they can be plotted in the Campbell diagram. For instance, that is the case of the forces resulting from the unbalance of the rotor, which can be modelled with frequency equal to the rotation speed Ω. If the shape of the force function is less regular, it can be split into harmonic terms using Fourier decomposition. The complete response of the system will be the sum of the responses of the system under those harmonic components.

Quite often, the relation between the frequency of the forcing function and Ω is of simple proportionality, and thus can be represented in the Campbell diagram by a straight line. If the proportion is one to one, the line $\omega = \Omega$ is graphed, and the excitation is said to be **synchronous**. The spin speeds Ω at which a forcing function has a frequency coinciding with a natural frequency of the system at that same Ω are known as **critical speeds**. They can be found in the diagram by looking for the intersections between natural and forcing frequencies.

As one may suspect, the critical speeds bring up **resonance**, and thus may be avoided as great displacements may be encountered. If that is the case, the rotor cannot operate near this speed without encountering strong vibrations and even catastrophic failure. However, not all critical speeds are equally dangerous, and that will depend on the main modes of the structure. Flexural critical speed are those linked with the **flexural** natural frequencies, and use to be particularly dangerous. They can be detected in the Campbell diagram by looking for the intersection with the line $\omega = \Omega$. In addition to these flexural speeds, torsional critical speeds can also be very dangerous.

The range starting from zero to the first critical speed is referred as subcritical range, and above it, supercritical range starts. In the past, it was thought that operation above the first critical speed was impossible, but it was later denied by De Laval, Stodola and many other rotordynamists. It is easy to mix up natural and critical speeds, mostly due that in many cases their value coincides. Even if that happens, the two physical phenomena are very different, particularly concerning the stressing of the rotor.

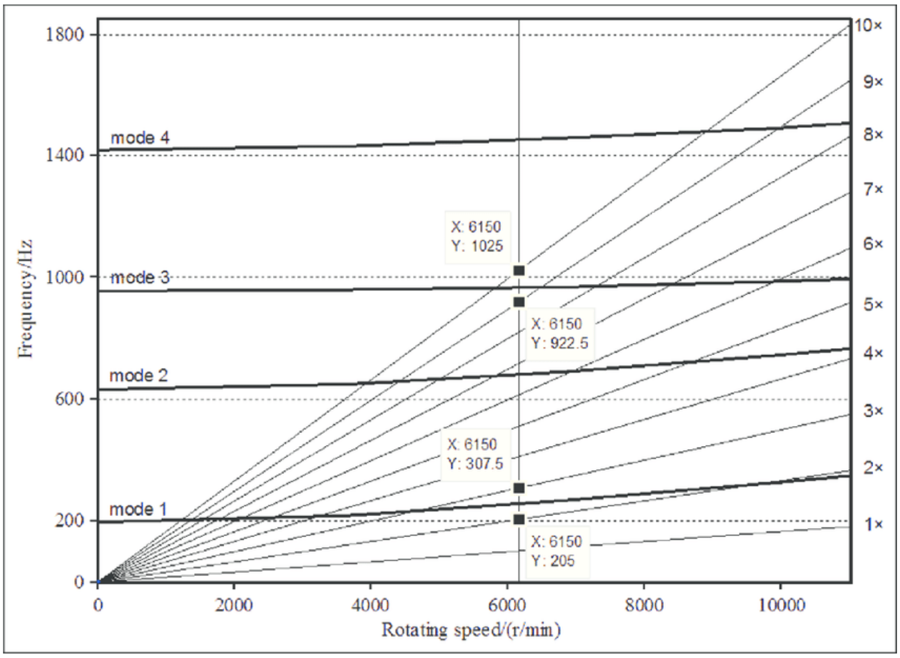


Figure 1.3: Example of Campbell diagram [6]

To illustrate a little bit this section, figure 1.3 shows the Campbell diagram of a compressor rotor blade. Bold curves represent natural frequencies of the structure, while the straight lines show the exciting frequencies. Each intersection between them is a resonance point. Here, rotation introduces an **stiffening** effect: natural frequencies increase with the spin speed.

It is important to remember that the concept of critical speeds and the Campbell diagram only apply to linear systems, although a more arbitrary definition of these speeds can be used in the case of nonlinear systems, referring the speed at which greater amplitudes are found.

Frequency and time domains

Linear rotordynamics can be analysed in the frequency domain, a widely used approach that brings up an insight on how the rotor behaves under different conditions. These kind of solutions allow the engineers to perform parametric design and optimization studies, as natural frequencies and vibration modes are easily obtained using this frame of analysis. When the system is linear and operating in steady-state condition, the most popular tool to analyse its behaviour is modal decomposition, which reduces the system of equations to an eigenvalue problem.

This is achieved, however, when the rotating structure is idealised. If the system is studied in the transient state, no frequency analysis can be achieved and only **numerical integration** in time is possible. Although integrating numerically the equations of motion is a widely used method nowadays to tackle complex problems, it only allows to solve very particular cases without bringing up an overall insight on the phenomena. Moreover, there is a high computational cost associated to this kind of numerical methods which is not present when performing a frequency analysis.

1.2.3. Nonlinear and nonstationary rotordynamics

The previous section presented a linearised model for a rotating structure. This is, however, only an idealisation as real rotors always deviate from the lineal model. Rotors are often designed to behave linearly in their nominal conditions. However, some sources of nonlinearity such as bearings, dampers and seals may disturb the system from its ideal behaviour. Typical approaches are to consider those elements as rigid bodies, or to linearise its behaviour, in order to simplify the resulting equations. This, however, leads to **unsatisfactory** approximations.

If the structure is highly nonlinear, sub- and super-harmonics play an important role, as well as chaotic oscillations. Chaos has been spotted in nonlinear rotor models, and numerical results indicate that they are present in turbojet engines. If the whirl amplitude grows in time, nonlinearities emerge until the system reaches a **limit cycle**.

As is to expect from nonlinear systems, sudden jumps from an equilibrium configuration to another may happen. Moreover, hysteresis may appear, making the spin-up of the system different from the deceleration, even if the speeds are the same. When the system to study becomes that complex, the only way to solve the equation is by means of numerical integration.

The previous linear analysis required from constant spin velocity Ω to be valid. It is, in fact, a common simplification in rotordynamics analysis. This model do not explains, however, the accelerating and decelerating transient states, or quick changes in the forcing functions. Even if the rotor is assumed to be linear, angular accelerations may introduce new terms in the matrices of equation 1.3 which will also become time-dependant. As a consequence, the system becomes complicated and the equations of motion are no longer solvable analytically. As an integrating numerical scheme will be required to solve the nonstationary system, there is little advantage in previously linearising the equation of motion.

1.2.4. The Jeffcott rotor

The analytical study of rotors dates back to the late nineteenth century, when Föppl, Stodola and Belluzzo described the behaviour of a simplified rotor. This model was deeply studied by Jeffcott (1919), and thus received its name. The model consists of a **point mass** attached to a massless shaft, being one of the simplest models used to describe the flexural behaviour of rotors. Yet, this is an **oversimplification** and one has to understand its limitations, although it may bring up qualitative insight into important phenomena of rotordynamics.

The model covered in this section considers an axisymmetrical fixed-free rotor, in which damping plays no role whatsoever. The restoring force is provided by an overall stiffness k, which will be considered the stiffness of the shaft. Let's consider an inertial frame xy and rotating noninertial frame $\varsigma\varrho$. In addition, small displacements are assumed, and the point mass is always to be contained in the xy plane. As in classic rotordynamics, the spin speed Ω is assumed to be constant.

Previously to the study of the Jeffcott rotor, let's consider an even simpler model consisting in a point mass constrained to move only in the axial direction ς. This is the kind of model **Rankine** used to develop his rotor analysis. The Lagrangian to the system yields to:

$$\mathfrak{L} = \frac{1}{2} m \left(\dot{\varsigma}^2 + \Omega^2 \varsigma^2 \right) - \frac{1}{2} k \varsigma^2 \tag{1.6}$$

And the equation of motion results in

$$m\ddot{\varsigma} + \left(k - m\Omega^2\right)\varsigma = 0 \tag{1.7}$$

This equations shows a decrease in the stiffness known as centrifugal softening, as the term multiplying the displacement is $k - m\Omega^2$, which is clearly smaller than k. Moreover, this system becomes unstable above the critical speed $\omega = \sqrt{(k/m)}$, prohibiting operation in the supercritical range.

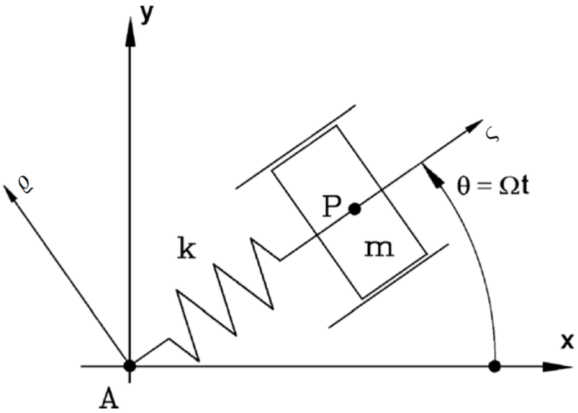


Figure 1.4: Rankine rotor model [3]

If the guide is removed, displacement in the transversal direction ϱ becomes possible, and a **Jeffcott** rotor is obtained. The equation of motion in the rotating coordinates results in

$$m\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{Bmatrix} \ddot{\varsigma} \\ \ddot{\varrho} \end{Bmatrix} - 2m\Omega\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{Bmatrix} \dot{\varsigma} \\ \dot{\varrho} \end{Bmatrix} + \begin{bmatrix} k - m\Omega^2 & 0 \\ 0 & k - m\Omega^2 \end{bmatrix}\begin{Bmatrix} \varsigma \\ \varrho \end{Bmatrix} = 0 \tag{1.8}$$

This model now takes into consideration **Coriolis** forces, which push the mass out of is axis, allowing self-centring and thus operation in the supercritical range. The equation of motion can be written in the inertial frame using a simple change of coordinates consisting in a rotation of angle Ωt. The equation of motion of the Jeffcott rotor in the inertial frame reads as

$$\begin{aligned} m\ddot{x} + kx = 0 \\ m\ddot{y} + ky = 0 \end{aligned} \tag{1.9}$$

The equation above predicts no softening effect nor instability in the inertial frame, and a whirl frequency $\omega_n = \pm\sqrt{k/m}$, equal to the flexural critical speed Ω_{cr}. For the case of a simply supported shaft, transverse stiffness k is equal to $\frac{48EI}{L^3}$ [7], where E stands for the Young's modulus, I for the second moment of area of the shaft cross-section and L for the span of the shaft. The above equation predicts that, if the initial conditions are such that the mass moves into circular orbits, the system is not affected by any kind of vibration but describes whirling motion instead. In the inertial frame, angular frequency may not be constant, but the period will be equal to $2\pi/\omega_n$.

The natural frequency of the Jeffcott rotor does not depend on Ω, and thus the Campbell diagram consists in horizontal straight lines. This is the simplest way of tackling the Jeffcott rotor, although more **realistic** variations of the model are widely used, taking into account forcing functions, damping, shaft bow, unbalance and out-of-plane displacements.

1.2.5. Gyroscopic effect

In the previous section, the Jeffcott model considered a rotor consisting in a point mass, and consequently, its moments of inertia were not taken into account. This is, however, a huge simplification. In reality, Campbell diagrams are not horizontal lines, but natural frequencies of bending modes depend on the spin speed. If one aims to account for this behaviour, has to consider the influence of **inertia moments**. Strictly speaking, the system has six degrees of freedom, and so six generalised coordinates should be defined in order to study its dynamic behaviour.

This behaviour was first noticed by Rayleigh, who identified the effect of the rotatory inertia of a disk mounted on a shaft. Stodola also considered this effect, and identified it as the responsible of giving rise to the split of natural frequencies. The study of critical speeds under the gyroscopic effect was carried by Green, and later studied using energy methods by Carnegie.

One of the simplest models follows the main assumptions of the Jeffcott rotor, but with a **rigid body** with nonvanishing moments of inertia instead of a point mass. In the undeformed position, one of the principal axes of inertia coincides with z. Principals moments of inertia will be referred as polar moment of inertia J_p, about the rotation axis, and transversal moment of inertia J_t, about any axes in the rotation plane. In matrix form:

$$\mathbf{J} = \begin{bmatrix} J_t & 0 & 0 \\ 0 & J_t & 0 \\ 0 & 0 & J_p \end{bmatrix} \tag{1.10}$$

In order to illustrate the problem, imagine a beam and its two first structural modes (figure 1.5). The first mode introduces a kinetic energy term, as the mass and the disk have been displaced by the deflection of the shaft. However, if the shaft is massless, there is no kinetic energy associated to the second mode, as the point mass has not changed its position. On the contrary, if the disk rotation given by the slope $(\partial v/\partial x)$ is taken into account, kinetic energy associated to the angular velocity emerges and has the following expression:

$$\frac{1}{2} J_T \left\{ \frac{d}{dt} \left(\frac{\partial v}{\partial x} \right) \right\}^2 \tag{1.11}$$

The above value of rotation energy is clearly zero if the beam is considered to be a point mass with no inertia, but when dealing with disks and rigid bodies, the inertia term cannot be neglected and neither can the rotation energy associated. This was a very known problem in static structures, known as rotatory inertia of beams. However, when the shaft rotates, the body acts like a spin top and **gyroscope**, having a significant effect in rotordynamics.

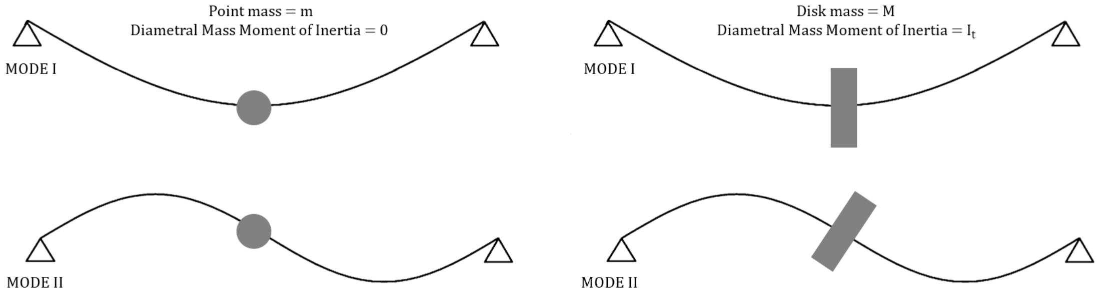

Point mass = m
Diametral Mass Moment of Inertia = 0
MODE I
MODE II
Disk mass = M
Diametral Mass Moment of Inertia = I_t
MODE I
MODE II

Figure 1.5: Massless shaft modes (lumped mass and disk models)

To understand the physics behind this phenomena, one has to remember Euler's equation for rigid bodies, which are clearly nonlinear with the angular velocities (see equation 1.2). Imagine a freely spinning disk (for instance, figure 1.6) with angular velocity Ω, rotating around an axis perpendicular to its non-deformed symmetry plane. Consider now that a whirling motion is introduced now around the x axis, due to any kind of perturbation or imbalance.

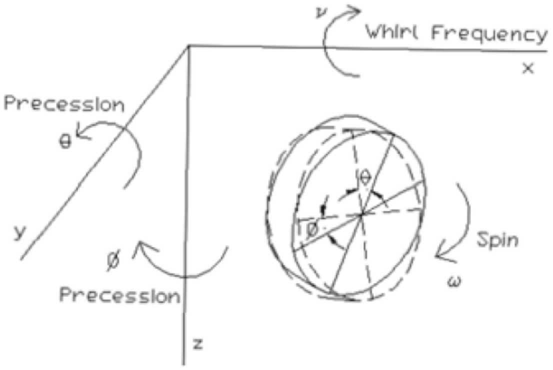


Figure 1.6: Free spinning disk [1]

Because of whirling, precessional motions linked to the shaft slopes at the disk centre are introduced. The disk then moves similar to a spin top, and the gyroscopic effect is introduced. Moreover, precession can happen in both xz and yz planes, leading to two different gyroscopic torques at which inertia torques ($J_T \cdot \ddot{\theta}_i$ and $J_T \cdot \ddot{\phi}_i$) are added. In this context, the moment relations across the disc i are written as

$$\begin{aligned} M_{y\,i} &= J_p \cdot \Omega \cdot \dot{\phi}_i + J_T \cdot \ddot{\theta}_i \\ M_{z\,i} &= -J_p \cdot \Omega \cdot \dot{\theta}_i + J_T \cdot \ddot{\phi}_i \end{aligned} \tag{1.12}$$

This relations have to be taken into account when modelling the dynamics of the system. In the model seen in equation 1.3, these effects are included in the gyroscopic matrix **G**. A further simplification can be achieved, **uncoupling** rotational and translational motions, leading only to four degrees of freedom. If that is the case, one can define the non-dimensional term $\delta = \frac{J_p}{J_t}$, and study the behaviour of the rotor as a function of δ.

The Jeffcott model, for instance, corresponds to $\delta = 0$, in which case curves in the Campbell diagram are horizontal lines. If $\delta > 1$, the rotor is disk-like, and as there is no intersection of the curves in the Campbell diagram with the line $\omega = \Omega$, no critical speed exist. On the contrary, if $\delta < 1$, the rotor is considered to be very long, and a critical speed linked with conical motion exists. Finally, if δ is exactly one, the rotor is spherical and although there is no intersection with the line $\omega = \Omega$, it tends asymptotically to it as Ω increases. Regarding the frequency of whirling, it decreases in absolute value as Ω increases for the case of backward whirling, and increases with an inclined asymptote $\omega = \delta\Omega$ in the case of forward whirling.

1.2.6. Centrifugal softening and stiffening

In previous sections, the concepts of centrifugal softening and stiffening have been used to refer the decrease and increase, respectively, of the natural frequencies of a rotating structure, compared with the natural frequencies of the same structure but standstill. While centrifugal stiffening is a very well-known effect, centrifugal softening is *"an alleged and elusive phenomena, and some doubts can be cast on the mathematical models causing it to appear"* [3]. A model of this type is used for instance in [1] (see page 285), trying to account for the softening effect in a solid rotor. The resulting equation is:

$$\mathbf{M}\ddot{\mathbf{q}} + (\mathbf{K} - \Omega^2\mathbf{M})\,\mathbf{q} = 0 \tag{1.13}$$

There is a term proportional to the mass matrix and to the square of the spin velocity, which is subtracted from **K** and thus natural frequencies wane. However, when flexibility and inertia of the disks and blades are accounted for, centrifugal **stiffening** plays a crucial role. It can be explained as the result of the stressing caused by tensile forces, which produces a virtual increase of stiffness proportional to the centrifugal force and thus to Ω^2. The simplest way to model a system accounting for both softening and stiffening effects is to use a $\mathbf{1\frac{1}{2}}$ **dimensional** approach, which consists in modelling shafts as beams and using axisymmetrical elements. The linearised homogeneous equation of motion [3] for the free vibration, taking into account axial, torsional, out-of-plane and in-plane bending behaviour reads as

$$\mathbf{M}\ddot{\mathbf{q}} + \Omega\mathbf{G}\dot{\mathbf{q}} + \left(\mathbf{K} + \mathbf{K}_\Omega\Omega^2 - \mathbf{M}_n\Omega^2\right)\mathbf{q} = \mathbf{0} \tag{1.14}$$

M, **G** and **K** are the mass, gyroscopic and stiffness matrices. $\mathbf{K}_\Omega$ is the centrifugal stiffening matrix, a geometric matrix whose effects are proportional to Ω^2. On the other hand, $\mathbf{M_n}$ is the centrifugal softening matrix, responsible for causing a decrease of the natural frequencies. The key is to assess whether natural frequencies actually decrease or not as Ω increases. To check that, it must be known whether the effect of $\mathbf{M_n}$ is stronger than $\mathbf{K}_\Omega$ or not.

The effect of $\mathbf{K}_\Omega$-$\mathbf{M_n}$ is dual: a stiffening effect due to tensile stresses caused by rotation and a softening due to the effects of the **noninertial** frame in which the equations are written. It can be demonstrated (see [3]) that that matrix is positive defined, and thus the overall net effect is of stiffening. The only models that, for its simplicity, are not able to capture the stiffening effect are the Rankine and Jeffcott rotors, which neither account for the gyroscopic effect. Moreover, the Rankine model is the only one that predicts a strong softening effect, which is known to lead to **incorrect** results.

1.3. Study of possibilities

The title of this thesis, *"Study of the vibration behaviour of rotatory blades using the finite element method"*, can lead to many different interpretations as it covers a broad subject. But some aspects are immovable: the main field of study is to be closely related to the subject of rotordynamics, and the tool of analysis has to rely somehow in the finite element method. Keeping this in mind, there are a few possibilities on how to face this study.

The first possibility consists in using an existing **commercial** solver that takes advantage of the FEM to simulate rotating structures.The main advantage of using an existing code is that makes complex and detailed studies possible. The equations behind the code are far more accurate than the ones that an undergraduate student may program, allowing a complete study of the dynamic behaviour of rotating structures. With this tool, **real** scenarios could be simulated, including wind blades and helicopter's rotors, obtaining thus accurate results.

This way of tackling the problem, however, does away with all the programming and FEM part of the project. The work of the student would be limited to the meshing of the structures and the analysis of the given results. Moreover, rotordynamic codes are not easy to use, and a considerable amount of time would be spent on learning how to use the program. It is not worth saying that those simulations would require for high computation capabilities, which are not cheap nor easy to obtain. Furthermore, rotordynamics is a topic widely covered by great investigators and big companies, so no new discovers would emerge from this study. For all the presented reasons, this will **not** be the approach used in this thesis.

A more plausible approach would consist in writing down a **code** to solve the rotordynamics equations, using an existing software (for instance, **GiD**) to carry the pre- and post-process of the numerical simulations. Among all the possible programming languages to use for the code, MATLAB seems to be a great choice, not only because of its simplicity and wide variety of functions that includes, but because it was the language used in the subject *Aerospace Computing Engineering*, taught by prof. *Hernández J.*, making code **reusability** possible.

Following this path, two different tracks are possible. The first one focuses on the modelling of an **accurate** rotating system (resembling equation A.20) using the finite element method. This equations need to be solved using numerical schemes, so again high computation capabilities would be needed. Moreover, posing this equations in terms of the FEM is not an easy task. As in the previous case, the solution of these equations and the vibration behaviour of a rotatory beam is already deeply understood, so nothing new would come to light.

That is why, instead of modelling the structure with complex and accurate equations, simpler models will be considered. The goal is not to get credible results, but to study on the **methods** used to solve the equations. For the cases where Ω is not constant, the equations cannot be solved in terms of modal decomposition and only numerical integration is possible. These numerical schemes take a lot of time and computation cost, and give only particular solutions of the system. The key of this approach is to develop a numeric method based on the **SVD** that has the advantage of the modal analysis of capturing global properties of the system —such as vibration modes and natural frequencies— when applied to rotors with angular acceleration and other nonlinearities.

1.4. Report outline

Regarding the written **organization** of the project, this thesis consists of three documents: the present report, an appendix including the report attachments (sections enumerated with capital Latin letters) and the project's budget. The report attachments contain extra information needed to achieve a deeper understanding of the whole project. Particular topics that could be from interest to the reader are presented there, as well as some figures related with the simulations carried out in chapters 6 and 7. The developed code is also attached in the appendix. As the present report often includes cross references with the attachments, the reader is invited to check **both** documents when reading the thesis.

Focusing on the report document, the chapters into which is divided are:

Chapter 2. The elastic model that describes the vibration behaviour of rotors is formulated. The strong form of the elastodynamic equation is obtained and analytically solved for the 1D stationary case.

Chapter 3. The weak form of the elastodynamic equations is obtained, posed in terms of the finite element method and formulated in the elemental frame. General concepts regarding the FEM methods are reviewed in this section.

Chapter 4. The general steps of every FEM simulation — preprocessing, solver and postprocessing — are reviewed. The 1D stationary case is solved using the FEM.

Chapter 5. Different methods to tackle the dynamic case are compared: modal decomposition analysis, numerical integration and singular value decomposition.

Chapter 6. The methods reviewed in chapter 5 are used to perform the structural analysis of rotating beams. In particular, the chapter focuses on the application of the SVD to recover predominant information of rotating structures.

Chapter 7. The dynamic model is further generalised, including aerodynamic forces and rigid-elastic coupling. The numeric solver is used to simulate the starting of an helicopter rotor.

Chapter 8. The developed work is reviewed and final conclusions are stated. Possible future tracks to continue with the present study are also discussed.

The presented outline may seem confusing for those with little experience in numerical methods. If that is the case, the reader should keep in mind the following scheme that encompasses the different parts of the **numerical tool** that is to be developed, as well as its chronological order:

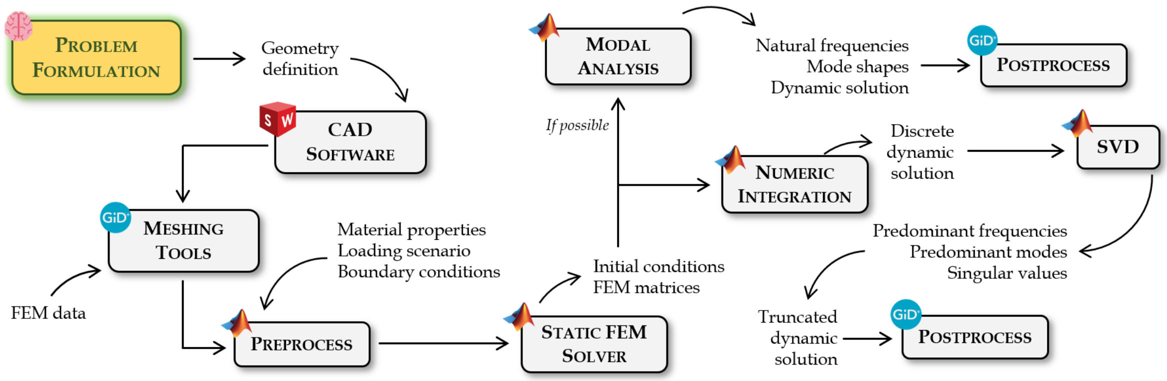


Figure 1.7: Overall numeric scheme [1]

[1] The used software is property of the following trademarks:

3D SOLIDWORKS™. 2019. Version 2018 SP4. Dassault Systèmes ®, Vélizy-Villacoublay, France.
GiD®. 2019. Official version 14.0.2. CIMNE ©, Barcelona, Spain.
MATLAB®. 2019. Version R2018a. The MathWorks, Inc., Natick, MA, USA.

2

Rotor modelling

The first step in the study of vibrations in rotating structures is to define an adequate model that describes its behaviour. In section 1.3, a justification on why a simple model is preferable to more accurate but complex equations is presented. The aim of this section is to keep going with this justification, presenting the chosen model and its limitations. The equations of motion will also be presented latter in this chapter.

2.1. Problem formulation and hypotheses

As it has already been stated in section 1.2, the study of rotordynamics can easily become complicated, as the understanding of the whole phenomena of rotating structures requires from a deep knowledge in several fields. However, an accurate study of rotatory structures is far from the scope of this project. Instead, more emphasis will be given to the **methods** used to solve the resulting equations, rather than on the modelling and simulation of the rotor. That is why the model will be defined under the following conditions and **hypotheses**:

- The axis of rotation of the rotor is known and **fixed**. Angular velocity and acceleration are known and given by the driving device.
- The problem is restricted to **small** displacements but great rotations induced by the driving device. Hence, it can be modelled using the linear elastic equations (A.1).
- The rotor will be modelled in terms of the finite element method using solid elements, being nodal displacements the degrees of freedom.
- The inertia of the rotor will not be taken into consideration, and thus no **gyroscopic effect** is expected.

The first point gives an important hint about the nature of the problem. In the inertial frame, rotating velocities and accelerations will be **known**. In other words, the inertial kinematic variables are not the unknown, which in fact is the torque needed to deliver the required power to maintain the given angular conditions. One has to take into account that, although the driving engine may introduce **large rotations**, strains will still be small in a rotating frame of reference static with respect the rotor.

The strongest **limitation** of the model is given by the forth point: no gyroscopic effect will be taken into consideration. This limits the correctness of the model when applied to structures with non-negligible inertia. The gyroscopic effect would induce rotations in different axes, breaking the first hypothesis. With this type of formulation, great part of the phenomena description is lost, as complex models such as the one of equation A.20 are difficult to model using classical formulations.

The above hypotheses will make the model inherently **inaccurate**. The resulting simulations will not describe the vibration behaviour of real rotor blades, although it is a great first step into the actual rotordynamic problem. The key of this approach is to make equations easy to pose, and thus expedite the programming of the model in terms of the FEM. Hopefully enough, this will allow a deeper study on the methods used to solve the dynamic system.

2.2. Frames of reference and rotation matrix

Previous to the posing of the equations describing the kinematics of the system, it is convenient to define the different frames of references that will be used: an inertial frame xyz, fixed to a static point O; and a noninertial rotating frame $\varsigma\varrho\varpi$ fixed to the rotor — thus sharing its angular velocity Ω and acceleration α. The planes defined by xy and $\varsigma\varrho$ are meant to be coincident, and thus z and ϖ will be the same axis (since both basis are positive definite). From now on, they both will be referred as z, the rotation axis.

To change from one coordinate system to another, one can define a **rotation matrix**, taking advantage of the fact that the angular displacement is known for every value of time. Following Euler's definition of rotation, it is stated that rotation in space can always be described by a rotation along a certain axis over a certain angle. In this case, the axis of rotation is z. One can, for instance, represent the rotation in a two-dimensional space, being $\mathbf{r}_i$ and $\mathbf{r}_f$ the vectors describing initial and final positions, respectively, of a given point. The rotation is represented in figure 2.1:

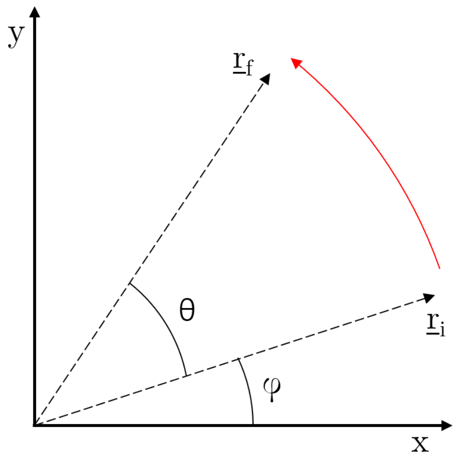


Figure 2.1: Vector rotation in the plane xy

$$\mathbf{r}_f = \begin{pmatrix} x_f \\ y_f \end{pmatrix} = \begin{pmatrix} R\cos(\theta+\varphi) \\ R\sin(\theta+\varphi) \end{pmatrix} = \begin{pmatrix} \overbrace{R\cos(\varphi)}^{x_i}\cos(\theta) - \overbrace{R\sin(\varphi)}^{y_i}\sin(\theta) \\ \underbrace{R\sin(\varphi)}_{y_i}\cos(\theta) - \underbrace{R\cos(\varphi)}_{x_i}\sin(\theta) \end{pmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \mathbf{Q}_{(\theta)}\,\mathbf{r}_i \tag{2.1}$$

Where $\mathbf{Q}_{(\theta)}$ stands for the rotation matrix, and it is satisfied that $\|\mathbf{r}_i\| = \left\|\mathbf{r}_f\right\| = R$.

It will be interesting, when posing the kinetic equations, to consider the cross product or **spin** of a vector t, represented with $\widetilde{t}$ and defined as the linear operator:

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \Rightarrow \widetilde{t} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \tag{2.2}$$

Using this operator, the rotation can be expressed in terms of the Rodrigues formula [8]:

$$\mathbf{Q}_{(\theta,\mathbf{n})} = \left[\cos\theta\,\mathrm{I} + (1-\cos\theta)\mathrm{n}\,\mathrm{n}^T + \sin\theta\,\tilde{\mathrm{n}}\right] \tag{2.3}$$

Which for the case $\mathbf{n} = \widehat{\mathbf{k}}$ gives the same result as in equation 2.1. What is interesting about the spin operator is the easiness to represent matrices derivatives. Defining the temporal derivative of a function as $\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} = \dot{\mathbf{f}}$, one can compute the temporal derivative of the three-dimensional rotation matrix using the spin operator:

$$\frac{\mathrm{d}\mathbf{Q}_{(\theta)}}{\mathrm{d}t} = \dot{\mathbf{Q}}_{(\theta)} = \begin{bmatrix} -\sin(\theta)\,\dot{\theta} & -\cos(\theta)\,\dot{\theta} & 0 \\ \cos(\theta)\,\dot{\theta} & -\sin(\theta)\,\dot{\theta} & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{Q}_{(\theta)} \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{Q}_{(\theta)}\,\tilde{\dot{\theta}} = \mathbf{Q}_{(\theta)}\,\widetilde{\Omega} \tag{2.4}$$

The above expression can be generalised to compute matrix derivatives of order n:

$$\frac{\mathrm{d}^{\mathrm{n}}\mathbf{Q}_{(\theta)}}{\mathrm{d}t^n} = \mathbf{Q}_{(\theta)}\,\widetilde{\Omega}^{\,n} \tag{2.5}$$

2.3. Kinematic equations

The kinematic description of the rotor is going to be quite simple. Imagine a particle of the rotor, represented by a **point mass** with no inertia associated. This point mass is subjected to a rotation given by the driving device. If the body were perfectly rigid, displacements in the rotating frame $\varsigma\varrho$ would be zero, and positions in the xy plane would be given by the driving device. This is, however, not the case. Displacements in all ς, ϱ and even z directions may occur.

To illustrate this effect, imagine a two-dimensional case in which elastic displacements are only possible in ς. Imagine the mass point starting from an initial position $\mathbf{r}_i$, which after a certain time holds a generic position $\mathbf{r}$. The total displacement $\mathbf{U}$ can be split in terms of rigid body $\mathbf{U}_{rb}$ and elastic $\mathbf{U}_{el}$ displacements:

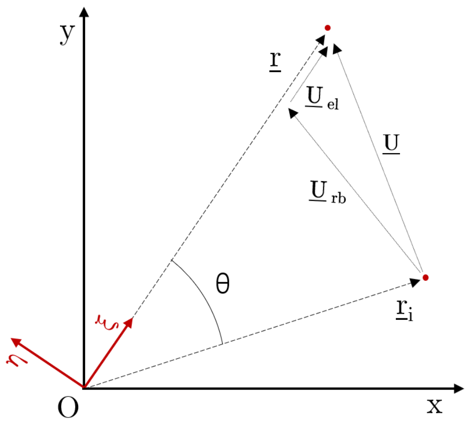
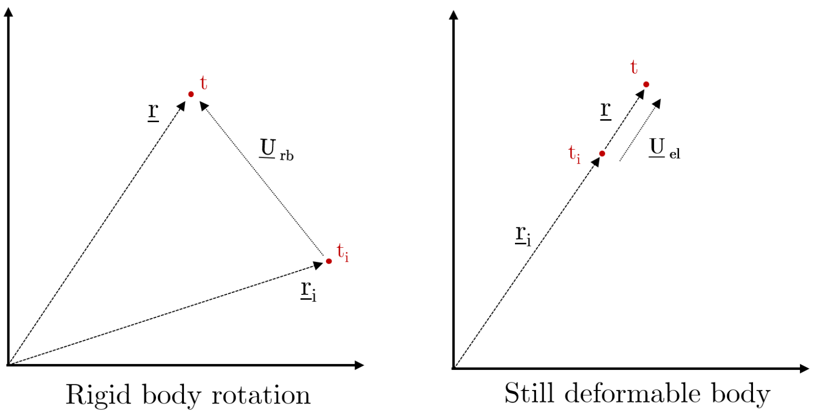


Figure 2.2: Displacement as composition of rotation and deformation

The equations describing this motion are easy to deduce. For simplicity, given a vector v, let's denote as $v_{(x)}$ the vector described in inertial coordinates, and $v_{(\varsigma)}$ the same vector but in the corotational coordinates. For instance, $\mathbf{r}_{(x)} = \left(x\,y\,z\right)^T$ and $\mathbf{r}_{(\varsigma)} = \left(\varsigma\,\varrho\,\varpi\right)^T$. It can be easily deduced from equation 2.2:

$$\mathbf{U}_{(x)} = \mathbf{r}_{(x)} - \mathbf{r}_{i\,(x)} = \mathbf{U}_{rb\,(x)} + \mathbf{U}_{el\,(x)} \tag{2.6}$$

The following figure helps on visualising the different displacements affecting the point mass, starting from a given time t_i until a generic t. The total displacement results from the combination of both motions. Note that, for simplification, only **axial** elastic deformations are represented:



Rigid body rotation

Still deformable body

Figure 2.3: Representation of the rigid body rotation and axial elastic deformation

Using the rotation matrix obtained in equation 2.1, one can develop the following relations regarding the **displacements** of the particle:

$$\mathbf{U}_{rb\,(x)} = \mathbf{Q}_{(\theta)}\,\mathbf{r}_i - \mathbf{r}_i = (\mathbf{Q}_{(\theta)} - \mathrm{I})\,\mathbf{r}_i\,, \quad \mathbf{U}_{el\,(x)} = \mathbf{Q}_{(\theta)}\,\mathbf{U}_{el\,(\varsigma)} \tag{2.7}$$

For the sake of notational simplification, let's simply denote the total displacement in the inertial frame $\mathbf{U}_{(x)}$ as $\mathbf{U}$, and the elastic deformations in the corotating frame $\mathbf{U}_{el\,(\varsigma)}$ as $\mathbf{u}$. With this notation, the equations describing the particle's position, velocity and acceleration are:

$$\mathbf{U} = \mathbf{Q}_{(\theta)}\,(\mathbf{r}_i + \mathbf{u}) - \mathbf{r}_i \tag{2.8}$$

$$\dot{\mathbf{U}} = \mathbf{Q}_{(\theta)}\left[\widetilde{\Omega}\,(\mathbf{r}_i + \mathbf{u}) + \dot{\mathbf{u}}\right] \tag{2.9}$$

$$\ddot{\mathbf{U}} = \mathbf{Q}_{(\theta)}\left[\underbrace{\widetilde{\Omega}^2\,(\mathbf{r}_i + \mathbf{u})}_{Centrifugal} + \underbrace{\widetilde{\alpha}\,(\mathbf{r}_i + \mathbf{u})}_{Tangential} + \overbrace{2\,\widetilde{\Omega}\,\dot{\mathbf{u}}}^{Coriolis} + \underbrace{\ddot{\mathbf{u}}}_{Relative}\right] \tag{2.10}$$

Where α stands the angular acceleration set by the driving device. Although until now, the figures accounted only for axial elastic deformations — that is, $\mathbf{u} = \left(u_\varsigma, 0, 0\right)^T$ — the fact is that the previous equations can be used to describe a general deformation in three dimensions: $\mathbf{u} = \left(u_\varsigma, u_\varrho, u_\varpi\right)^T$. From the previous equations, Ω and α are imposed by the driving device, while $\mathbf{r}_i$ only depends on the initial condition. The elastic kinematic **unknowns** are, then, $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$.

2.4. Elasticity notation

Previous to the posing of the equilibrium equations, some definitions regarding the used notation should be clarified. Let n_d denote the number of space **dimensions** of the problem under consideration, and let $\mho \subset \mathbb{R}^{n_d}$ be an open set with piecewise smooth boundary Γ. A general point, denoted by $\boldsymbol{x}$, is identified with its position vector emanating from the **origin**. Tensors and vectors will be described in terms of Cartesian components. In indicial notation, differentiation is denoted by:

$$\frac{\partial}{\partial x_i} = \partial_{x_i} = \partial_i\,, \quad \partial_{ii} = \sum_{i=1}^{n_d} \partial_{ii} \tag{2.11}$$

This time, a general form of the elastic problem is to be developed. The displacement field, referred as $\mathfrak{u}$, pretends to represent displacement in a generic case, and should not be confused with the elastic deformations in the corotational frame $\mathbf{u}$, presented in the previous section. Regarding the domain boundary, we shall assume that it does not change it time and admits the following decomposition:

$$\Gamma = \Gamma^i_{\mathfrak{u}} \cup \Gamma^i_{\sigma}\,, \;\; \Gamma^i_{\mathfrak{u}} \cap \Gamma^i_{\sigma} = \varnothing \tag{2.12}$$

With the previous decomposition, two different **boundary conditions** can be stated. Regarding the prescribed displacements, the **Dirichlet** boundary conditions $\mathfrak{u}_i = \bar{\mathfrak{u}}^i$ are defined on $\Gamma^i_{\mathfrak{u}}$. Note that prescribed displacements may change in time, thus $\bar{\mathfrak{u}}_i : \Gamma^i_u \times]0,T[\to \mathbb{R}$, where $]0,T[$ is the open time interval of length $T > 0$. On the other hand, regarding the prescribed tractions, the **Neumann** boundary conditions $\bar{\mathfrak{t}}^i = \sigma_{ij} n_j$ are defined on Γ^i_{σ}, where again $\bar{\mathfrak{t}}_i : \Gamma^i_{\sigma} \times]0,T[\to \mathbb{R}$.

To complete the definition of the boundary value problem, **initial conditions** shall be given. Since the governing equation will involve accelerations and thus be second order in time, both initial displacements and velocities must be specified:

$$\mathfrak{u}_{i\,(\boldsymbol{x},0)} = \mathfrak{u}^0_{i\,(\boldsymbol{x})}\,, \quad \dot{\mathfrak{u}}_{i\,(\boldsymbol{x},0)} = \dot{\mathfrak{u}}^0_{i\,(\boldsymbol{x})} \tag{2.13}$$

As stated in the previous section, the key is to find the displacements, which are represented by a vector field $\mathfrak{u} : \mho \to \mathbb{R}^{n_d}$. Using infinitesimal theory (see A.1), the deformation state at a given point $\boldsymbol{x} \in \mho$ is characterised by the infinitesimal **strain tensor**, which is defined to be the symmetric part of the displacement gradients:

$$\varepsilon_{ij} := \left[\nabla^s \mathfrak{u}\right]_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.14}$$

For linear elastostatic systems, the relationship between the Cauchy stress tensor and the strain tensor is given in equation A.6. In the dynamic case, however, the stress at a given time is not only proportional to the strain but also to its derivative (linear **viscoelasticity**). In indicial form:

$$\sigma_{ij} = \mathbf{C}_{ijkl}\varepsilon_{kl} + \overline{\mathbf{D}}_{ijkl}\dot{\varepsilon}_{kl} \tag{2.15}$$

Where $\overline{\mathbf{D}}$ is the tensor of viscoelasticity coefficients. It is often assumed to be proportional to the elasticity tensor, $\overline{\mathbf{D}} = \beta\mathbf{C}$. For the dynamic case, inertia forces — proportional to acceleration — have to be taken into account, and will be treated as body forces per unit volume. Additionally, a damping term proportional to velocity is also to be included:

$$\overbrace{f_i}^{static} \to \overbrace{f_i - \rho\ddot{\mathfrak{u}}_i - \overline{\mu}\,\dot{\mathfrak{u}}_i}^{dynamic} \tag{2.16}$$

Where $\rho : \overline{\underset{\sim}{\Omega}} \to \mathbb{R}$ stands for the material density, and $\overline{\mu}$ is a damping parameter usually defined as $\overline{\mu} = \overline{\alpha}\rho$, where $\overline{\alpha}$ is known as damping coefficients, often considered to be constant.

2.5. Strong form of the elastodynamic equation

Supposing the rotor as perfectly elastic, the equation describing the dynamic behaviour is given by Newton's second law for **continuum** media, which in indicial notation reads as:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \left(\boldsymbol{f}_i - \overline{\mu}\,\dot{\mathfrak{u}}_i\right) = \rho \ddot{\mathfrak{u}}_i \tag{2.17}$$

But to solve the previous equation, the boundary conditions and the elastic model defined in the previous section are required. The **strong form** of the elastodynamic boundary value problem is then formulated as:

$$\tag{2.18}$$

$$\boxed{\begin{array}{c}
\text{Given } \boldsymbol{f}_i, \overline{\mathfrak{u}}^i,\ \overline{\mathbf{t}}^i,\ \mathfrak{u}_i^0 \text{ and } \dot{\mathfrak{u}}_i^0\ \text{, find } \mathfrak{u}_i : \overline{\mho} \times]0,T[\rightarrow \mathbb{R} \text{ such that} \\
\frac{\partial \sigma_{ij}}{\partial x_j} + \left(\boldsymbol{f}_i - \overline{\mu}\,\dot{\mathfrak{u}}_i\right) = \rho\, \ddot{\mathfrak{u}}_i \ \text{ in } \mho \times]0,T[\\
\mathfrak{u}_i = \overline{\mathfrak{u}}^i \ \text{ on } \Gamma^i_{\mathfrak{u}} \times]0,T[\\
\sigma_{ij} n_j = \overline{\mathbf{t}}^i \ \text{ on } \Gamma^i_{\sigma} \times]0,T[\\
\mathfrak{u}_{i\,(\boldsymbol{x},0)} = \mathfrak{u}^0_{i\,(\boldsymbol{x})} \ \text{ and } \ \dot{\mathfrak{u}}_{i\,(\boldsymbol{x},0)} = \dot{\mathfrak{u}}^0_{i\,(\boldsymbol{x})} \ \text{ with } \ \boldsymbol{x} \in \mho \\
\text{where } \ \sigma_{ij} = \mathbf{C}_{ijkl}\varepsilon_{kl} + \overline{\mathbf{D}}_{ijkl}\dot{\varepsilon}_{kl}
\end{array}}$$

The previous system has no closed form **solution**, and an analytical solution is only found for simple domains . For more complex geometries and boundary conditions, approximate methods have to be used. There is no need to point out that the effect of rotation does nothing but adds complexity to the equations. In the inertial frame, equation 2.17 yields to the following expression when accounting for the acceleration induced by rotation:

$$\nabla \boldsymbol{\sigma}_{\boldsymbol{x}} + \mathbf{Q}_{(\theta)} \left(\mathbf{f} - \overline{\mu}\,\dot{\mathbf{u}}\right) = \rho\, \mathbf{Q}_{(\theta)} \left[\widetilde{\Omega}^2 \left(\mathbf{r}_i + \mathbf{u}\right) + \widetilde{\alpha} \left(\mathbf{r}_i + \mathbf{u}\right) + 2\,\widetilde{\Omega}\,\dot{\mathbf{u}} + \ddot{\mathbf{u}}\right] \tag{2.19}$$

2.5.1. Transformation of coordinates

One of the problems of the above equation is that the **stress field** is difficult to pose in terms of the inertial frame, as the elasticity tensor is only known in the corotating coordinates. Using the rotation matrix, stresses can be transformed from one basis to another:

$$\sigma_{\boldsymbol{x}} = \ \mathbf{Q}_{(\theta)}\ \sigma_{\boldsymbol{\varsigma}}\ {\mathbf{Q}_{(\theta)}}^T \tag{2.20}$$

Upon some manipulations, the above expression can be rephrased in **Voigt**'s notation as follows:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xz} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos^2\theta & \sin^2\theta & 0 & 0 & 0 & -\sin 2\theta \\ \sin^2\theta & \cos^2\theta & 0 & 0 & 0 & \sin 2\theta \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0.5(\sin 2\theta) & -0.5(\sin 2\theta) & 0 & 0 & 0 & \cos 2\theta \end{bmatrix}}_{\mathbf{T}_{(\theta)}} \begin{bmatrix} \sigma_\varsigma \\ \sigma_\varrho \\ \sigma_\varpi \\ \tau_{\varrho\varpi} \\ \tau_{\varsigma\varpi} \\ \tau_{\varsigma\varrho} \end{bmatrix} \tag{2.21}$$

Notice that $\mathbf{T}^{-1}_{(\theta)} = \mathbf{T}_{(-\theta)}$.

Then, the relation between the stresses in the inertial $\boldsymbol{\sigma}_{\boldsymbol{x}}$ and in the corotational frame $\boldsymbol{\sigma}_{\boldsymbol{\varsigma}}$, and strains in the inertial $\boldsymbol{\varepsilon}_{\boldsymbol{x}}$ and in the corotational frame $\boldsymbol{\varepsilon}_{\boldsymbol{\varsigma}}$ in Voigt notation is given by:

$$\boldsymbol{\sigma}_{\boldsymbol{x}} = \mathbf{T}_{(\theta)}\, \boldsymbol{\sigma}_{\boldsymbol{\varsigma}}\,, \quad \boldsymbol{\varepsilon}_{\boldsymbol{x}} = \mathbf{T}^{-T}_{(\theta)}\, \boldsymbol{\varepsilon}_{\boldsymbol{\varsigma}} \tag{2.22}$$

An analogous expression can be deduced regarding the elasticity tensor:

$$\mathbf{C}_{\boldsymbol{x}} = \mathbf{T}_{(\theta)}\ \mathbf{C}_{\boldsymbol{\varsigma}}\ \mathbf{T}^{T}_{(\theta)} \tag{2.23}$$

Having defined these relations, equation 2.19 can be rewritten in terms of the known elasticity matrix $\mathbf{C}_{(\varsigma)}$:

$$\nabla\left(\mathbf{T}_{(\theta)}\, \boldsymbol{\sigma}_{\boldsymbol{\varsigma}}\right) + \mathbf{Q}_{(\theta)} \left(\mathbf{f} - \overline{\mu}\,\dot{\mathbf{u}}\right) = \rho\ \mathbf{Q}_{(\theta)} \left[\widetilde{\Omega}^2\,(\mathbf{r}_i + \mathbf{u}) + \widetilde{\alpha}\,(\mathbf{r}_i + \mathbf{u}) + 2\,\widetilde{\Omega}\,\dot{\mathbf{u}} + \ddot{\mathbf{u}}\right] \tag{2.24}$$

If the dynamics of the system are studied in the **noninertial frame**, the above equation reduces to

$$\nabla\boldsymbol{\sigma}_{\boldsymbol{\varsigma}} + \left(\mathbf{f} - \overline{\mu}\,\dot{\mathbf{u}}\right) = \rho\ \left[\widetilde{\Omega}^2\,(\mathbf{r}_i + \mathbf{u}) + \widetilde{\alpha}\,(\mathbf{r}_i + \mathbf{u}) + 2\,\widetilde{\Omega}\,\dot{\mathbf{u}} + \ddot{\mathbf{u}}\right] \tag{2.25}$$

2.5.2. One-dimensional stationary solution

One of the simplest cases that can be tackled analytically is a one-dimensional stationary approach. In other words, it means that elastic deformations **u** are only present in the **axial** direction ς, while the rotation is considered to be stationary with constant angular velocity Ω. Moreover, no vibration phenomena is taken into consideration, and so $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are neglected. That is to imagine the rotor as a sequence of one-dimensional bars, which can only experience pure traction and compression. For this case where $n_d = 1$, neglecting damping and external forces, equation 2.25 yields to

$$\frac{d A_{(\varsigma)}\, \sigma_\varsigma}{\partial\varsigma} = -\rho\, A_{(\varsigma)}\, \widetilde{\Omega}^2\,(\mathbf{r}_i + \mathbf{u}) \ \Rightarrow\ A\,\frac{d\left(\mathbf{C}_{\varsigma\varsigma}\,\varepsilon_\varsigma\right)}{d\varsigma} = -\rho\, A\,\Omega^2\,(\varsigma + u_\varsigma) \tag{2.26}$$

Identifying the coefficient $\mathbf{C}_{\varsigma\varsigma}$ as the Young's Modulus, considered independent on the axial position, and considering constant section A, the above equation can be rewritten as:

$$E\,A\,\frac{d^2 u_\varsigma}{d\varsigma^2} = -\rho\, A\,\Omega^2\,(\varsigma + u_\varsigma) \ \Rightarrow\ \frac{d^2 u_\varsigma}{d\varsigma^2} = -\frac{1}{E}\,\rho\,\Omega^2\,(\varsigma + u_\varsigma) \tag{2.27}$$

Where ρ stands for the density of the material and E is the Young's Modulus. The above equation describes the rotation of a 1D structure under constant angular velocity. In the corotational frame, the deformation can be interpreted as the result of applying an axially distributed force, whose modulus is proportional to the radial position (figure 2.4). Note that the displacement field will not depend on the area of the beam.

The resulting equation can be identified as a second order ordinary differential equation (**ODE**). Thus, two boundary conditions will be required. Supposing a **fixed-free** structure, the displacement at the root is to be zero, and the same happens with the stress at the free end, where $\varsigma = R$:

$$u\Big|_{\varsigma=0} = 0\,, \quad \frac{du_\varsigma}{d\varsigma}\Big|_{\varsigma=R} = 0 \tag{2.28}$$

To simplify the solution, let's denote $k = \sqrt{\frac{\rho\,\Omega^2}{E}}$.

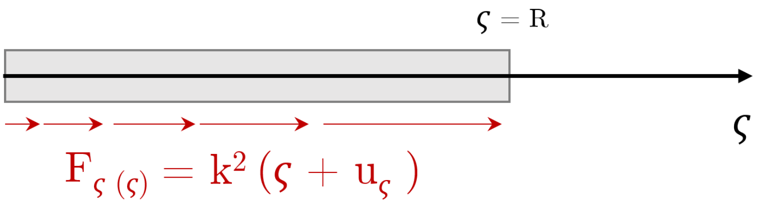

$\varsigma = R$
ς
$F_{\varsigma\,(\varsigma)} = k^2\,(\varsigma + u_\varsigma\,)$

Figure 2.4: 1D static problem interpreted as an axially loaded static bar in the rotating frame

The solutions of the ODE are:

$$u_{\varsigma\,(\varsigma)} = \frac{\sin(k\varsigma)}{k\cos(kR)} - \varsigma\,, \quad \sigma_{\varsigma\,(\varsigma)} = E\left(\frac{\cos(k\varsigma)}{\cos(kR)} - 1\right) \tag{2.29}$$

Imagine now an helicopter's rotor, with a constant cross-section and $4\,m$ radius rotating at $60\,rad/s$. Let's be idealistic, and suppose that the blade can only deform in the axial direction. Moreover, let's imagine that the conditions are perfectly **ideal** and no source of vibration exists. In that case, the previous equations could be used to compute the deformations and stresses across the beam. For an aluminium beam ($E = 70\,GPa$, $\rho = 2700\,kg/m^3$), the obtained results are showed in figure 2.5, displaying an overall elongation of the blade of $3\,mm$:

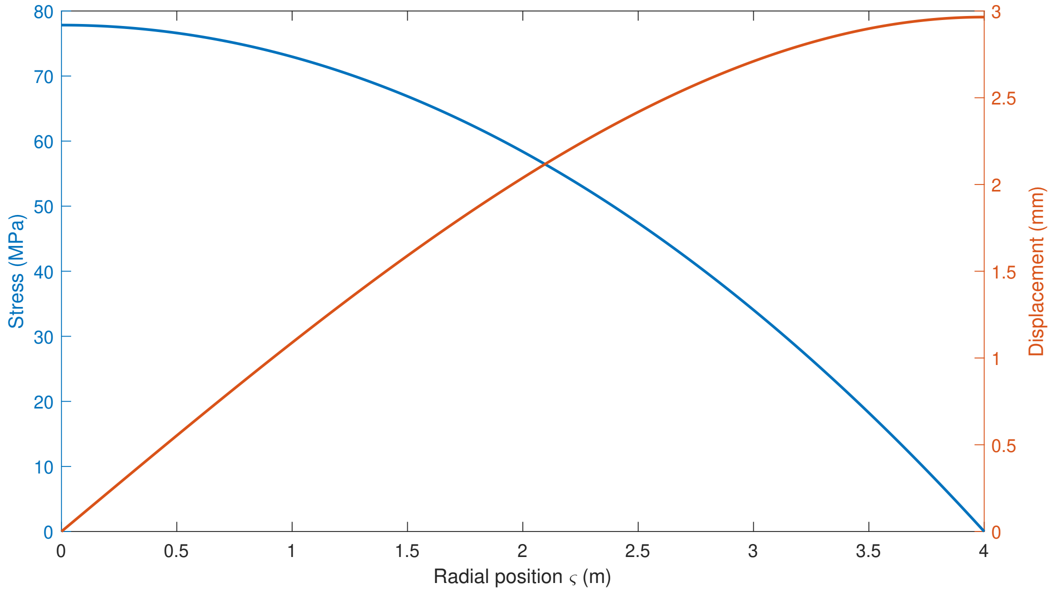


Figure 2.5: Analytic solution for an axially loaded bar due to centrifugal forces

For sure, this model cannot be applied to real rotors. However, it is an exact solution to equation 2.25, and will be used latter on this report to **validate** the developed approximated methods.

3

FEM model

Once the strong form of the elastodynamic equation has been posed, is now time to transform it in terms of the **finite element method**, in such a way that it can be solved using this computation technique. The aim of the present chapter is to pose the weak form of the equation and give a glimpse on how to solve it. It is taken for granted that the reader has basic notions on the topic of FEM, and thus this chapter will not explain the very basics of the method.

3.1. Weak form of the elastodynamic equation

For the sake of simplicity, let's take advantage of the formulation presented in the previous chapter while developing the strong form of the problem. To start with, the **test** and **trial** functions will be defined. Those are continuous with square-integrable derivative functions, defined over the problem boundary $\overline{\mho}$, with specific boundary values:

$$\begin{aligned} \textbf{Test functions } \mathcal{V} &= \left\{ v_i : \overline{\mho} \rightarrow \mathbb{R},\ v_i = 0 \text{ on } \Gamma_u^i \right\} \\ \textbf{Trial functions } \mathcal{S} &= \left\{ u_i : \overline{\mho} \rightarrow \mathbb{R},\ u_i = \overline{u}_i \text{ on } \Gamma_u^i \right\} \end{aligned} \tag{3.1}$$

Denoting as f_{ij} a C_1 function, the Gauss' divergence theorem reads as:

$$\int_{\mho} \frac{\partial f_{ij}}{\partial x_j} d\mho = \int_{\Gamma} f_{ij}\, n_j d\Gamma \tag{3.2}$$

The weak form is obtained by integrating over the domain the **product** of the balance equation (see eq. 2.17) in with the test functions, and by employing the **Divergence Theorem** to shift the derivatives on the stresses σ_{ij} to the test functions v_i. For simplicity, let's denote $(f_i - \overline{\mu}\,\dot{u}_i - \rho\,\ddot{u}_i)$ as $\underset{\sim}{f_i}$.

$$\int_{\mho} v_i \left(\frac{\partial \sigma_{ij}}{\partial x_j} + \underset{\sim}{f_i} \right) d\mho = 0 \ \Rightarrow\ \int_{\mho} v_i\, \frac{\partial \sigma_{ij}}{\partial x_j} d\mho + \int_{\mho} v_i\, \underset{\sim}{f_i} d\mho = 0 \tag{3.3}$$

Integrating by parts and applying equation 3.2 to the first term:

$$\int_{\mho} v_i \frac{\partial \sigma_{ij}}{\partial x_j} d\mho = \int_{\Gamma} v_i \left(\sigma_{ij}\, n_j \right) d\Gamma - \int_{\mho} \frac{\partial v_i}{\partial x_j}\, \sigma_{ij} d\mho \tag{3.4}$$

The first term on the right side of the preceding equation can be expressed as the sum of the integrals over Γ_u^i and Γ_σ^i. Taking into account that $v_i = 0$ on Γ_u^i, and $\sigma_{ij} n_j = \overline{t}^i$ on Γ_σ^i :

$$\int_{\Gamma} v_i \left(\sigma_{ij}\, n_j\right) d\Gamma = \sum_{i=1}^{n_d} \int_{\Gamma_u^i} v_i \left(\sigma_{ij}\, n_j\right) d\Gamma + \sum_{i=1}^{n_d} \int_{\Gamma_\sigma^i} v_i \left(\sigma_{ij}\, n_j\right) d\Gamma = \sum_{i=1}^{n_d} \int_{\Gamma_\sigma^i} v_i\, \bar{t}^{\,i} d\Gamma \tag{3.5}$$

Introducing the latter into equation 3.3:

$$\sum_{i=1}^{n_d} \int_{\Gamma_\sigma^i} v_i\, \bar{t}^{\,i} d\Gamma - \int_{\mho} \frac{\partial v_i}{\partial x_j}\, \sigma_{ij}\, d\mho + \int_{\mho} v_i\, \underset{\sim}{f_i}\, d\mho = 0 \tag{3.6}$$

As $\sigma_{ij} = \sigma_{ji}$, the derivatives of the test function can be expressed in terms of the **symmetric gradient operator**: $\frac{\partial v_i}{\partial x_j}\sigma_{ij} = [\nabla^s v]_{ij}\sigma_{ij}$. It will be convenient to express the problem in **Voigt's notation** (see A.1.2). Thus, the latter expression $[\nabla^s v]_{ij}\sigma_{ij}$ transforms into $(\nabla^s \boldsymbol{v})^T \boldsymbol{\sigma}$.

Taking into account the initial conditions defined in the previous chapter, the **weak form** of the elastodynamic problem using Voigt's notation is written as:

(3.7)

Given $\mathbf{f}$, $\bar{\boldsymbol{\mathfrak{u}}}^i$, $\bar{\mathbf{t}}^i$, $\boldsymbol{\mathfrak{u}}^0$ and $\dot{\boldsymbol{\mathfrak{u}}}^0$, find $\boldsymbol{\mathfrak{u}}_{(t)} \in \mathcal{S}$, $t \in [0, T]$ such that

$$\int_{\mho} \mathbf{v}^T \rho\, \ddot{\boldsymbol{\mathfrak{u}}}\, d\mho + \int_{\mho} \mathbf{v}^T \bar{\mu}\, \dot{\boldsymbol{\mathfrak{u}}}\, d\mho + \int_{\mho} \nabla^s \mathbf{v}^T\, \overline{D}\, \nabla^s \dot{\boldsymbol{\mathfrak{u}}}\, d\mho +$$

$$+ \int_{\mho} \nabla^s \mathbf{v}^T\, \mathbf{C}\, \nabla^s \boldsymbol{\mathfrak{u}}\, d\mho \;= \int_{\mho} \mathbf{v}^T \boldsymbol{f}\, d\mho + \sum_{i=1}^{n_d} \int_{\Gamma_\sigma^i} v_i \bar{t}^{\,i} d\Gamma\,,\ \forall \mathbf{v} \in \mathcal{V}$$

$$\boldsymbol{\mathfrak{u}} = \bar{\boldsymbol{\mathfrak{u}}}^i \ \text{ on }\ \Gamma_{\boldsymbol{\mathfrak{u}}}^i \times [0, T]$$

$$\boldsymbol{\mathfrak{u}}_{(\boldsymbol{x},0)} = \boldsymbol{\mathfrak{u}}^0_{(\boldsymbol{x})} \ \text{ and }\ \dot{\boldsymbol{\mathfrak{u}}}_{(\boldsymbol{x},0)} = \dot{\boldsymbol{\mathfrak{u}}}^0_{(\boldsymbol{x})} \ \text{ with }\ \boldsymbol{x} \in \mho$$

The above expression is known as the variational form of the elastodynamic problem, and can be seen as the **balance** of internal and external virtual works, an equilibrium between stresses and inertia, viscous, body and traction forces. The term $\nabla^s \mathbf{v}^T$ can be thought as virtual strains, while $\mathbf{v}^T$ as virtual displacements. The major difference with the strong formulation is that the weaker form of the problem demands less **smoothness** of the solution, involving only firsts derivatives of the test functions.

The formulation of the weak form of the elastodynamic problem is only the first step for the finite element method to be fully developed. The following sections will introduce the required formulation to tackle the problem in terms of the FEM, allowing its programming.

3.2. Global formulation

The key concept of the FEM lies in the domain decomposition into a certain number of **elements**, finite-sized subdomains. Each element has a given number of *nodes*, which will depend on the element type. The important fact here is to know that the FEM only computes values of the variable at the nodes. Then, the values of non-nodal points are approximated by **interpolation** of the nodal values. The FEM equations are formulated in such a way that continuity of the field variables is ensured. However, interelement continuity of the gradients of the variable does not often exist. The latter is a problem affecting some variables of interest such as strains and stresses, and is solved by increasing the number of elements used.

Following with the concept of domain decomposition, it is said that the domain $\overline{\mho}$ is **discretised** into element domains $\overline{\mho}^e$. The geometric shape resulting from the domain division is known as mesh. If the domain can be thought as a polygon, and n_{el} is the total number of elements in which the domain is split, the decomposition can be written as

$$\overline{\mho} = \overline{\mho}^1 \cup \overline{\mho}^2 \cdots \cup \overline{\mho}^{n_{el}} = \bigcup_{e=1}^{n_{el}} \overline{\mho}^e \tag{3.8}$$

If n_{pt} is the total number of nodes of the domain, then the FEM has to compute the field variable (in this case, displacement) in each of those nodes. For n_d dimensions, each nodal displacement will be a n_d-sized vector. We can define a global vector of displacements $\mathbf{d} \in \mathbb{R}^{n_d n_{pt}}$ and variations $\mathbf{c} \in \mathbb{R}^{n_d n_{pt}}$:

$$\mathbf{d} := \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_{n_{pt}} \end{bmatrix}, \quad \mathbf{c} := \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_{n_{pt}} \end{bmatrix} \tag{3.9}$$

It is important to notice that not all the components of the previous vectors are unknown: $\mathbf{r} \subset \{1, 2 \ldots n_d n_{pt}\}$ is defined as the set of **constrained** degrees of freedom (DOF), along which displacement is known. Thus, $\mathbf{d}_r = \bar{\mathrm{u}}$ and $\mathbf{c}_r = 0$. On the other hand, $\mathbf{l}$ is known as the set of **unconstrained** degrees of freedom, the global DOFs along which displacement is unknown.

As it has been stated in the first paragraph of this section, the solution along non-nodal points is interpolated using the nodal values. The interpolation functions are known as **shape functions**, and in the **Galerkin** method the same functions are used to interpolate both shape and trial functions. Let's define the continuous and with piecewise continuous derivative scalar shape function $N_A : \overline{\mho} \to \mathbb{R}$, where $1 \leq A \leq n_{pt}$.

In the FEM, nodal shape functions N_A are defined in such a way that their value is **1** at node A, and **0** at every other node. In problems tackling with vector fields, a matrix of nodal shape functions is defined. Using the same function to interpolate displacements in all dimensions, the nodal matrix of shape functions for a three-dimensional case is:

$$\mathbf{N}_A := N_A \mathbf{I} = N_A \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{N}_A \in \mathbb{R}^{n_d \times n_d} \tag{3.10}$$

A global matrix of shape functions $\mathbf{N} \to \mathbb{R}^{n_d \times n_d n_{pt}}$ can be defined, involving all the nodal matrices: $\mathbf{N} := \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \cdots & \mathbf{N}_{n_{pt}} \end{bmatrix}$. In the same line, a global matrix $\mathbf{B}$ involving the gradient of the shape functions is defined: $\mathbf{B} = \nabla^s \mathbf{N}$.

The next step is to construct finite-dimensional **approximations** of the test and trial functions, denoted as $\mathcal{V}^h$ and $\mathcal{S}^h$, respectively. If one thinks of these approximations as subsets of $\mathcal{V}$ and $\mathcal{S}$, and if $u^h \subset \mathcal{S}^h$ and $v^h \subset \mathcal{V}^h$, then the approximated functions will meet the boundary conditions defined in 3.1. The **finite** dimensional space of test and trial functions is defined as the space including all the functions of the form

$$\begin{aligned} \mathcal{V}^h &\to \mathbf{v}^h_{(\boldsymbol{x})} = \mathbf{N}_{(\boldsymbol{x})} \, \mathbf{c}, \text{ with } \mathbf{c}_{\mathrm{r}} = 0 \\ \mathcal{S}^h &\to \mathbf{u}^h_{(\boldsymbol{x},\, t)} = \mathbf{N}_{(\boldsymbol{x})} \, \mathbf{d}_{(t)}, \text{ with } \mathbf{d}_{\mathrm{r}(t)} = \bar{\mathbf{u}}_{(t)} \end{aligned} \tag{3.11}$$

In a similar way, velocity $\dot{\mathfrak{u}}$ and acceleration $\ddot{\mathfrak{u}}$ are approximated as $\dot{\mathfrak{u}}^h = \mathbf{N}_{(\boldsymbol{x})}\,\dot{\mathbf{d}}_{(t)}$ and $\ddot{\mathfrak{u}}^h = \mathbf{N}_{(\boldsymbol{x})}\,\ddot{\mathbf{d}}_{(t)}$, respectively. With all the previous definitions being formulated, the approximated weak form of the problem can be posed:

$$\boxed{\begin{gathered}
\text{Given } \mathbf{f},\ \overline{\mathfrak{u}}^{\,i},\ \bar{\mathbf{t}}^{\,i},\ \mathfrak{u}^0 \text{ and } \dot{\mathfrak{u}}^0\ \text{, find } \mathfrak{u}^h_{(t)} \in \mathcal{S}^h,\ \ t \in [0,T]\ \text{ such that}\\[2ex]
\int_{\mho} \mathbf{v}^{h^T} \rho\, \ddot{\mathfrak{u}}^h d\mho + \int_{\mho} \mathbf{v}^{h^T} \overline{\mu}\, \dot{\mathfrak{u}}^h\, d\mho + \int_{\mho} \nabla^s \mathbf{v}^{h^T} \overline{D} \nabla^s \dot{\mathfrak{u}}^h\, d\mho +\\
+ \int_{\mho} \nabla^s \mathbf{v}^{h^T} \mathbf{C} \nabla^s \mathfrak{u}^h\, d\mho\ = \int_{\mho} \mathbf{v}^{h^T} \boldsymbol{f}\, d\mho + \sum_{i=1}^{n_d} \int_{\Gamma^i_\sigma} v^h_i \bar{t}^{\,i}\, d\Gamma,\ \forall \mathbf{v} \in \mathcal{V}\\[2ex]
\mathfrak{u}^h = \overline{\mathfrak{u}}^{\,i}\ \text{ on }\ \Gamma^i_{\mathfrak{u}} \times [0,T]\\
\mathfrak{u}^h_{(\boldsymbol{x},0)} = \mathfrak{u}^0_{(\boldsymbol{x})}\ \text{ and }\ \dot{\mathfrak{u}}^h_{(\boldsymbol{x},0)} = \dot{\mathfrak{u}}^0_{(\boldsymbol{x})}\ \text{ with }\ \boldsymbol{x} \in \mho
\end{gathered}} \tag{3.12}$$

3.2.1. Approximated rotating elastodynamic model

Until now, rotation has not been taken into account when modelling the weak form of the elastodynamic problem. Introducing the **kinematic equations** obtained in section 2.3, substituting the approximations presented in equation 3.11 and using the algebraic property $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$, the above expression can be written in the noninertial frame as

$$\begin{aligned}
\mathbf{c}^T \Bigg\{ & \overbrace{\left(\int_{\mho} \mathbf{N}^T \rho\, \mathbf{N}\, d\mho\right)}^{\mathbf{M}} \ddot{\mathbf{d}} + \overbrace{\left(\underbrace{\int_{\mho} \mathbf{N}^T \overline{\mu}\, \mathbf{N}\, d\mho + \int_{\mho} \mathbf{B}^T \overline{\mathbf{D}}\, \mathbf{B}\, d\mho}_{\mathbf{D}_{static}} + \underbrace{2 \int_{\mho} \mathbf{N}^T \rho\, \widetilde{\Omega}\, \mathbf{N}\, d\mho}_{\mathbf{D}_{rot}} \right)}^{\mathbf{D}} \dot{\mathbf{d}} +\\
& + \overbrace{\left(\underbrace{\int_{\mho} \mathbf{B}^T \mathbf{C}\, \mathbf{B}\, d\mho}_{\mathbf{K}_{static}} + \underbrace{\int_{\mho} \mathbf{N}^T \rho\, (\widetilde{\Omega}^2 + \widetilde{\alpha})\, \mathbf{N}\, d\mho}_{\mathbf{K}_{rot}} \right)}^{\mathbf{K}} \mathbf{d} -\\
& - \overbrace{\left(\underbrace{\int_{\mho} \mathbf{N}^T \boldsymbol{f}\, d\mho + \sum_{i=1}^{n_d} \int_{\Gamma^i_\sigma} \overline{\mathbf{N}}^T_i\, \bar{t}^{\,i}\, d\Gamma}_{\mathbf{F}_{static}} - \underbrace{\int_{\mho} \mathbf{N}^T \rho\, (\widetilde{\Omega}^2 + \widetilde{\alpha})\, \mathbf{r}_i\, d\mho}_{\mathbf{F}_{rot}} \right)}^{\mathbf{F}} \Bigg\} = 0
\end{aligned} \tag{3.13}$$

In compact form:

$$\mathbf{c}^T \left[\mathbf{M}\ddot{\mathbf{d}} + \mathbf{D}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} - \mathbf{F}\right] = 0$$

$$\mathbf{c}^T \left[\mathbf{M}\ddot{\mathbf{d}} + (\mathbf{D}_{static} + \mathbf{D}_{rot})\,\dot{\mathbf{d}} + (\mathbf{K}_{static} + \mathbf{K}_{rot})\,\mathbf{d} - (\mathbf{F}_{static} - \mathbf{F}_{rot})\right] = 0 \tag{3.14}$$

Where the subscript *"static"* refers to those matrices that are general in the FEM when applied to static structures, whereas those matrices with the subscript *"rot"* are particular of this specific case of rotating structure. Recall that in the above equation, **d** stands for the approximate elastic displacement in the rotative frame — $\mathbf{u}^h_{(\boldsymbol{x},t)} = \mathbf{N}_{(x)}\,\mathbf{d}_{(t)}$.

The obtained matrices are very common in physical systems. **M** is known as mass matrix, **D** as damping matrix, **K** as stiffness matrix and **F** as force vector. Quite often, $\overline{\mathbf{D}}$ and $\overline{\boldsymbol{\mu}}$ are not known. A common approach is to suppose that the system is governed by the **Rayleigh damping**, which assumes $\overline{\boldsymbol{\mu}} = \overline{\alpha}\rho$ and $\overline{\mathbf{D}} = \overline{\beta}\mathbf{C}$, where $\overline{\alpha}$ and $\overline{\beta}$ are assumed to be constant parameters. Thus, the static damping matrix in the noninertial frame can be expressed as:

$$\mathbf{D}_{static} = \overline{\alpha}\mathbf{M} + \overline{\beta}\mathbf{K}_{static} \tag{3.15}$$

It is important to notice that, in the case where the angular velocity Ω is **constant**, the stiffness and damping matrices and the force vector due to rotation can be written as:

$$\mathbf{K}_{rot} = \int_{\mho} \mathbf{N}^T \rho\, \widetilde{\Omega}^2\, \mathbf{N}\, d\mho = -\Omega^2 \int_{\mho} \mathbf{N}^T \rho \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{N}\, d\mho$$

$$\mathbf{D}_{rot} = 2\int_{\mho} \mathbf{N}^T \rho\, \widetilde{\Omega}\, \mathbf{N}\, d\mho = 2\Omega \int_{\mho} \mathbf{N}^T \rho \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{N}\, d\mho$$

$$\mathbf{F}_{rot} = \int_{\mho} \mathbf{N}^T \rho\, (\widetilde{\Omega}^2 + \widetilde{\alpha})\, \mathbf{r}_i\, d\mho = -\Omega^2 \int_{\mho} \mathbf{N}^T \rho \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{r}_i\, d\mho \tag{3.16}$$

In this case, all the above expressions are constant in time. The key fact here is that, for the two-dimensional case, $\mathbf{K}_{rot}$ is proportional to the mass matrix by a factor of $-\Omega^2$. Then:

$$\mathbf{K} = \mathbf{K}_{static} - \Omega^2\mathbf{M} \tag{3.17}$$

This is a **softening effect** induced by the non-inertial character of the frame reference where the equations are posed. This behaviour has already been reviewed in the introductory part of this report (see section 1.2.6). This softening causes a reduction in the natural frequencies of the structure, compared to those of the static case. However, this effect is **rarely seen** in actual rotors. In reality, softening is a phenomena eclipsed by the so-called stiffening effect. The problem is that simple rotor models do not account for the latter, and thus the overall effect is of softening [3].

For sure, equation 3.14 can be posed in the **inertial** frame of reference using the rotation matrix and applying the transformation of stresses from equation 2.23:

$$\mathbf{M}_{inertial} = \int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \rho\, \mathbf{N}\, d\mho$$

$$\mathbf{D}_{inertial} = \int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \overline{\mu}\, \mathbf{N}\, d\mho + \int_{\mho} \mathbf{B}^T \mathbf{T}_{(\theta)}\, \overline{\mathbf{D}}\, \mathbf{T}^T_{(\theta)}\, \mathbf{B}\, d\mho + 2\int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \rho\, \widetilde{\Omega}\, \mathbf{N}\, d\mho$$

$$\mathbf{K}_{inertial} = \int_{\mho} \mathbf{B}^T \mathbf{T}_{(\theta)}\, \mathbf{C}\, \mathbf{T}^T_{(\theta)}\, \mathbf{B}\, d\mho + \int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \rho\, (\widetilde{\Omega}^2 + \widetilde{\alpha})\, \mathbf{N}\, d\mho$$

$$\mathbf{F}_{inertial} = \int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \boldsymbol{f} d\mho + \sum_{i=1}^{n_d} \int_{\Gamma^i_\sigma} \mathbf{N}_i^T\, \mathbf{T}_{(\theta)}\, \overline{t}^i d\Gamma - \int_{\mho} \mathbf{N}^T \mathbf{Q}_{(\theta)}\, \rho\, (\widetilde{\Omega}^2 + \widetilde{\alpha})\, \mathbf{r}_i\, d\mho \tag{3.18}$$

3.2.2. Block matrix decomposition

The above equations cannot be solved unless boundary values of displacement are specified. Moreover, the involved matrices do not admit inverse, so the only way to tackle the system is to break the presented matrices into **block matrices**. Remember that **r** stands for the set of DOFs where displacement is prescribed, whereas **l** holds for the set of DOFs where the displacement field is to be computed. Hence, naming $\mathbf{J}_{rl}$ the matrix made-up by the elements forming the intersection of the **r** rows and **l** columns of **J**, equation 3.14 is split into:

$$\begin{bmatrix}\mathbf{c_l}^T & \mathbf{c_r}^T\end{bmatrix}\left(\begin{bmatrix}\mathbf{M_{ll}} & \mathbf{M_{lr}}\\ \mathbf{M_{rl}} & \mathbf{M_{rr}}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{d}}_\mathbf{l}\\ \ddot{\mathbf{d}}_\mathbf{r}\end{bmatrix}+\begin{bmatrix}\mathbf{D_{ll}} & \mathbf{D_{lr}}\\ \mathbf{D_{rl}} & \mathbf{D_{rr}}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{d}}_\mathbf{l}\\ \dot{\mathbf{d}}_\mathbf{r}\end{bmatrix}+\begin{bmatrix}\mathbf{K_{ll}} & \mathbf{K_{lr}}\\ \mathbf{K_{rl}} & \mathbf{K_{rr}}\end{bmatrix}\begin{bmatrix}\mathbf{d_l}\\ \mathbf{d_r}\end{bmatrix}-\begin{bmatrix}\mathbf{F_l}\\ \mathbf{F_r}+\mathbf{R}\end{bmatrix}\right)=0 \tag{3.19}$$

Where **R** stands for the elastic reactions at the nodes with prescribed displacement, which are unknown. The above expression can be thought as a system of two equations:

$$\begin{aligned}\mathbf{c}_\mathbf{l}^T\left(\mathbf{M_{ll}}\ddot{\mathbf{d}}_\mathbf{l} + \mathbf{M_{lr}}\ddot{\mathbf{d}}_\mathbf{r} + \mathbf{D_{ll}}\dot{\mathbf{d}}_\mathbf{l} + \mathbf{D_{lr}}\dot{\mathbf{d}}_\mathbf{r} + \mathbf{K_{ll}}\mathbf{d_l} + \mathbf{K_{lr}}\mathbf{d_r} - \mathbf{F_l}\right) = 0\\ \mathbf{c}_\mathbf{r}^T\left(\mathbf{M_{rl}}\ddot{\mathbf{d}}_\mathbf{l} + \mathbf{M_{rr}}\ddot{\mathbf{d}}_\mathbf{r} + \mathbf{D_{rl}}\dot{\mathbf{d}}_\mathbf{l} + \mathbf{D_{rr}}\dot{\mathbf{d}}_\mathbf{r} + \mathbf{K_{rl}}\mathbf{d_l} + \mathbf{K_{rr}}\mathbf{d_r} - \mathbf{F_r} - \mathbf{R}\right) = 0\end{aligned} \tag{3.20}$$

The above equations are fulfilled for all $\mathbf{c}^T$ if and only if the terms multiplying both $\mathbf{c}_\mathbf{l}^T$ and $\mathbf{c}_\mathbf{r}^T$ vanish. Solving for the unknown displacements and reactions, and substituting $\ddot{\mathbf{d}}_\mathbf{r}$, $\dot{\mathbf{d}}_\mathbf{r}$ and $\mathbf{d_r}$ by the corresponding boundary values $\overline{\ddot{\mathbf{u}}}$, $\overline{\dot{\mathbf{u}}}$ and $\overline{\mathbf{u}}$, respectively, the resulting equations are:

$$\begin{aligned}\mathbf{M_{ll}}\ddot{\mathbf{d}}_\mathbf{l} + \mathbf{D_{ll}}\dot{\mathbf{d}}_\mathbf{l} + \mathbf{K_{ll}}\mathbf{d_l} = \mathbf{F_l} - (\mathbf{M_{lr}}\overline{\ddot{\mathbf{u}}} + \mathbf{D_{lr}}\overline{\dot{\mathbf{u}}} + \mathbf{K_{lr}}\overline{\mathbf{u}})\\ \mathbf{R} = \mathbf{M_{rl}}\ddot{\mathbf{d}}_\mathbf{l} + \mathbf{D_{rl}}\dot{\mathbf{d}}_\mathbf{l} + \mathbf{K_{rl}}\mathbf{d_l} + \mathbf{M_{rr}}\overline{\ddot{\mathbf{u}}} + \mathbf{D_{rr}}\overline{\dot{\mathbf{u}}} + \mathbf{K_{rr}}\overline{\mathbf{u}} - \mathbf{F_r}\end{aligned} \tag{3.21}$$

3.2.3. Reaction computation

Notice that the reactions can only be found once the displacements in the unconstrained degrees of freedom are known. If the equations are posed in the noninertial frame of reference, the obtained value of reaction is the expected from a static structure under the effect of a certain deformation. To account for the total reaction, the **rigid body** moment associated to the rotation motion is to be known:

$$\mathbf{W}_{rb} = \mathbf{J_z}\,\alpha \tag{3.22}$$

Where $\mathbf{W}_{rb}$ stands for the reaction torque associated to the rigid body motion, $\mathbf{J_z}$ is the inertia moment around the rotation axis and α is the angular acceleration. If this reaction torque is added to the torque generated by the elastic reactions in the noninertial frame **R**, the overall torque is obtained. In order to find the value of $\mathbf{W}_{rb}$, it will be necessary to compute $\mathbf{J_z}$, which fortunately can be posed in terms of the FEM. Defining the rigid body modes $\mathcal{R}$ in three dimensions:

$$\mathcal{R} = \begin{bmatrix}1 & 0 & 0 & 0 & \Delta z_1 & -\Delta y_1\\ 0 & 1 & 0 & -\Delta z_1 & 0 & \Delta x_1\\ 0 & 0 & 1 & \Delta y_1 & -\Delta x_1 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & 0 & 0 & 0 & \Delta z_{n_{pt}} & -\Delta y_{n_{pt}}\\ 0 & 1 & 0 & -\Delta z_{n_{pt}} & 0 & \Delta x_{n_{pt}}\\ 0 & 0 & 1 & \Delta y_{n_{pt}} & -\Delta x_{n_{pt}} & 0\end{bmatrix} = \begin{bmatrix}\mathbf{I} & -\Delta\tilde{\mathbf{r}}\\ \vdots & \vdots\end{bmatrix} \tag{3.23}$$

Where $\mathcal{R}$ is a matrix with 6 columns, corresponding to the **rigid body degrees of freedom**: three translations and three rotations. In the above expressions, the first three columns are the translation modes, and can be represented by the identity matrix **I**. On the other hand, the last three columns are the rotation modes, and are computed as the **spin** of the distance with respect to the rotation axis $-\Delta\widetilde{\mathbf{r}}$. The rigid body modes matrix has as many rows as degrees of freedom, which means that the sequence presented above is repeated for each node. In two-dimensional problems, only one rotational and two translational modes exist.

The key fact is that $\mathcal{R}$ can be used to normalise the mass matrix $\mathbf{M}$ in such a way that information about the structure is obtained. For a generic two-dimensional case it is found that:

$$\underbrace{\mathcal{R}^T}_{3\times n_{pt}}\underbrace{\mathbf{M}}_{n_{pt}\times n_{pt}}\underbrace{\mathcal{R}}_{n_{pt}\times 3}=\begin{bmatrix}\mathbf{m} & 0 & 0\\ 0 & \mathbf{m} & 0\\ 0 & 0 & \mathbf{J_z}\end{bmatrix} \tag{3.24}$$

Where $\mathbf{m}$ stands for the total mass of the structure, and $\mathbf{I_z}$ is the inertia moment around the rotation axis. These properties may seem to be trivial to compute, but determining either the mass or the inertia can be a difficult task if the structure is not defined by a simple geometry. Moreover, when the geometry is simple and those values can be computed analytically, they can be used to asses whether $\mathbf{M}$ has been computed correctly by comparing the analytic values of mass and inertia with those obtained using the above numerical expression.

Once the inertia is known, the torque $\mathbf{W}_{rb}$ associated to the rigid body motion can be computed. Then, the reaction torque due to elastic reactions $\mathbf{W}_{el}$ is to be computed by multiplying each term of $\mathbf{R}$ by the distance with respect to the rotation axis. For a generic case, the **elastic reaction torque** is computed in terms of each of the components of $\mathbf{R}$:

$$\mathbf{w}_{el}=\begin{bmatrix}0 & -\Delta z_1 & \Delta y_1\\ \Delta z_1 & 0 & -\Delta x_1\\ -\Delta y_1 & \Delta x_1 & 0\\ \vdots & \vdots & \vdots\\ 0 & -\Delta z_{n_{pt}} & \Delta y_{n_{pt}}\\ \Delta z_{n_{pt}} & 0 & -\Delta x_{n_{pt}}\\ -\Delta y_{n_{pt}} & \Delta x_{n_{pt}} & 0\end{bmatrix}\begin{pmatrix}R_{x\,1}\\ R_{y\,2}\\ R_{z\,3}\\ \vdots\\ R_{x\,n_{pt}}\\ R_{y\,n_{pt}}\\ R_{z\,n_{pt}}\end{pmatrix}=\begin{bmatrix}\Delta\widetilde{\mathbf{r}}\\ \vdots\end{bmatrix}\mathbf{R} \tag{3.25}$$

The total reaction torque $\mathbf{W}_{el}$ is an n_d vector computed by adding the contributions of $\mathbf{w}_{el}$ for every spatial direction. Finally, the total torque is:

$$\mathbf{T}=\mathbf{W}_{rb}+\mathbf{W}_{el} \tag{3.26}$$

When the kinematics of the problem are known — which is the case — one of the the most important unknowns to determine are the total forces and moments acting on the structure. The value of $\mathbf{T}$ gives a hint on how much **work** needs to be injected into the blade to ensure rotation at the given angular speed and acceleration.

3.3. Local formulation

The integrals presented in the previous sections cannot be computed at once unless the geometry of the domain is tremendously simple, which is not the case. The grace of the FEM is that breaks the domain into smaller elements with well defined geometric properties, where the previous integrals are **approximated** using Gauss Quadrature, allowing computer implementation . In order to achieve this goal, a new **local frame** within a single element $\mho^e$ is defined. Let's assume the model consists of n_{el} elements, and take e as the variable index for the elements $1\leq e\leq n_{el}$.

Let's introduce $\boldsymbol{x}^e \in \overline{\mho}^{\boldsymbol{e}}$, the vector of global or **physical** coordinates in the local frame, containing the coordinates of a given element **e**. Following this local approach, let's define $\boldsymbol{\xi} = (\xi\ \eta\ \zeta\ )^T \in \overline{\mho}_{\boldsymbol{\xi}}$ as the vector of local or element coordinates in the **parent** domain. In order to visualise the mapping from the parent domain to the physical domain using isoparametric linear elements, let's consider a simple 1D case. The transformation from one domain to another is represented in the following figure:

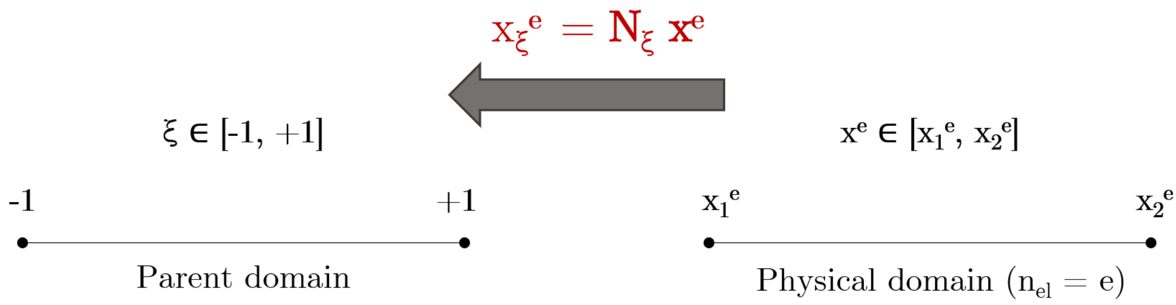

$\mathrm{x}_{\xi}{}^{\mathrm{e}} = \mathrm{N}_{\xi}\ \mathrm{x}^{\mathrm{e}}$
$\xi \in [-1, +1]$
$\mathrm{x}^{\mathrm{e}} \in [\mathrm{x}_1{}^{\mathrm{e}}, \mathrm{x}_2{}^{\mathrm{e}}]$
-1
+1
$\mathrm{x}_1{}^{\mathrm{e}}$
$\mathrm{x}_2{}^{\mathrm{e}}$
Parent domain
Physical domain ($\mathrm{n}_{\mathrm{el}} = \mathrm{e}$)

Figure 3.1: Representation of the mapping from the physical to the parent domain for a linear 1D element

The same concept can be applied to 2D and 3D elements. For a generic case, the mapping is constructed using the same shape functions employed in the interpolation of $\mathbf{u}^h$ (isoparametric elements). Defining n^e as the number of nodes of the element e, the mapping is expressed as:

$$\boldsymbol{x}^e_{(\xi)} = \begin{bmatrix} x_1^e & x_2^e & \cdots & x_{n_e}^e \\ y_1^e & y_2^e & \cdots & y_{n_e}^e \\ z_1^e & z_2^e & \cdots & z_{n_e}^e \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ \vdots \\ N_{n_e}^e \end{bmatrix} = \mathbf{X}^e \mathcal{N}^{e\ T}_{(\xi)} \tag{3.27}$$

Were $\mathbf{X}^e$ stands for the nodal coordinates in the physical domain, and $\mathcal{N}^e_{(\xi)} \in \mathbb{R}^{1\times n^e}$ is the matrix of shape functions for scalar valued fields. In a similar way, one may define the matrix of shape functions for vector valued fields $\mathbf{N}^e = [N_1^e\,\mathbf{I} \quad N_2^e\,\mathbf{I} \quad \cdots \quad N_{n_e}^e\,\mathbf{I}] \in \mathbb{R}^{n_d \times n_d\, n^e}$, where $\mathbf{I}$ is the $n_d \times n_d$ identity matrix.

The mapping can be interpreted as a **change of coordinates**, and thus it is a transformation with an associated Jacobian matrix $\mathbf{J}^e$. It allows the differential mapping between physical and parent domain, and vice versa:

$$\begin{aligned} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} &= \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} \quad \Rightarrow \quad d\boldsymbol{x} = \mathbf{J}^e d\boldsymbol{\xi} \\ \begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} &= \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad \Rightarrow \quad d\boldsymbol{\xi} = \mathbf{J}^{e-1} d\boldsymbol{x} \end{aligned} \tag{3.28}$$

Taking a closer look to the elemental Jacobian matrix, one founds that it can be expressed as

$$\mathbf{J}^e = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \xi} & \frac{\partial}{\partial \eta} & \frac{\partial}{\partial \zeta} \end{bmatrix} = \boldsymbol{x}\,\tilde{\nabla}^T_{\boldsymbol{\xi}} = (\mathbf{X}^e\,\mathcal{N}^{e\ T}_{(\xi)})\tilde{\nabla}^T_{\boldsymbol{\xi}} = \mathbf{X}^e\,(\tilde{\nabla}^T_{\boldsymbol{\xi}}\mathcal{N}^{e\ T}_{(\xi)}) = \mathbf{X}^e\,\tilde{\mathbf{B}}^{e^T}_{\xi} \tag{3.29}$$

Where $\tilde{\nabla}_{\boldsymbol{\xi}}$ is the **gradient operator** for scalar-valued functions in element coordinates and $\tilde{\mathbf{B}}^e_\xi$ is the matrix of the gradient of $\mathcal{N}^e$ for scalar-valued functions in the parent domain. The gradient operator for scalar-valued functions in both parent and physical domains ($\tilde{\nabla}_{\boldsymbol{\xi}}$ and $\tilde{\nabla}$) are linked through the transpose of the inverse of the Jacobian matrix:

$$\tilde{\nabla} = \mathbf{J}^{e-T}\,\tilde{\nabla}_{\boldsymbol{\xi}} \;\Rightarrow\; \tilde{\mathbf{B}}^e = \tilde{\nabla}\mathcal{N}^e = \mathbf{J}^{e-T}\left(\tilde{\nabla}_\xi \mathcal{N}^e\right) = \mathbf{J}^{e-T}\,\tilde{\mathbf{B}}^e_\xi \tag{3.30}$$

In the above expression, $\tilde{\mathbf{B}}^e \in \mathbb{R}^{n_d \times n^e}$ is the matrix of gradient of shape functions for scalar valued functions in global coordinates. In a 3D problem:

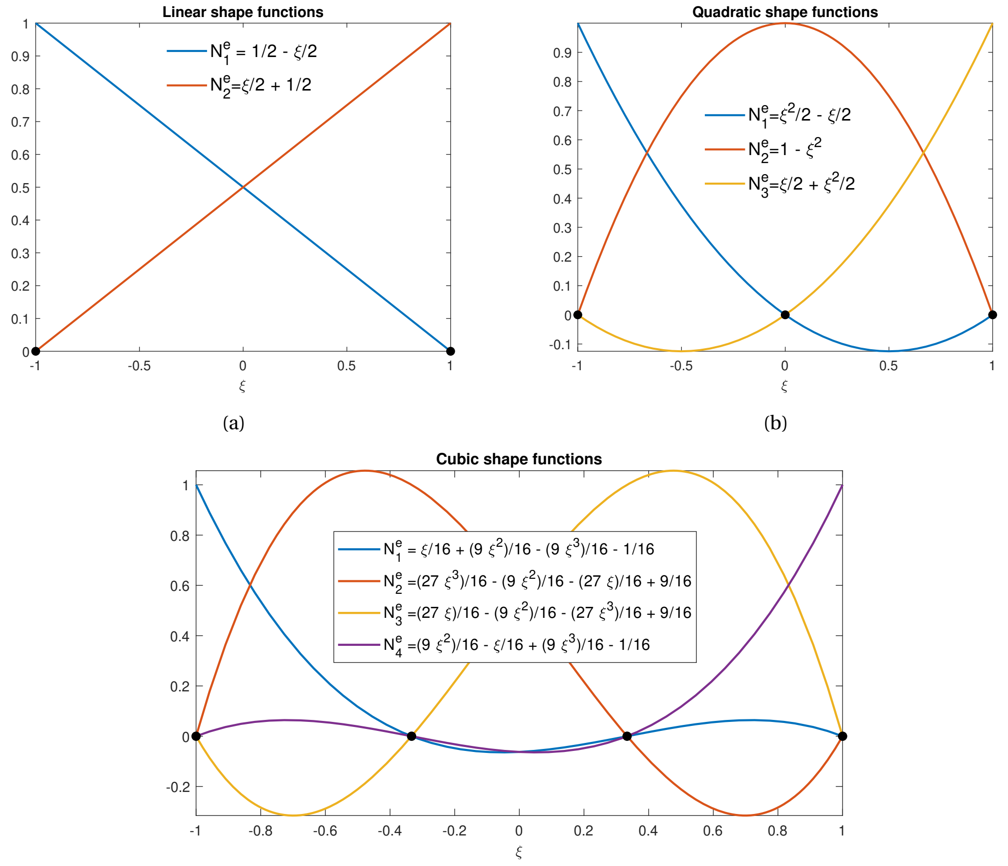
$$\tilde{\mathbf{B}}^e = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial x} & \cdots & \frac{\partial N_{ne}^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial y} & \cdots & \frac{\partial N_{ne}^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & \frac{\partial N_2^e}{\partial z} & \cdots & \frac{\partial N_{ne}^e}{\partial z} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{B}}_1^e & \tilde{\mathbf{B}}_2^e & \cdots & \tilde{\mathbf{B}}_{n^e}^e \end{bmatrix} \tag{3.31}$$

However, the displacement field to be computed is a **vector field**, so a matrix of symmetric gradient of shape functions for vector-valued functions in the physical domain $\mathbf{B}^e$ is to be defined. For a 3D problem:

$$\mathbf{B}^e = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & 0 & \cdots & \frac{\partial N_{ne}^e}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & \cdots & 0 & \frac{\partial N_{ne}^e}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1^e}{\partial z} & \cdots & 0 & 0 & \frac{\partial N_{ne}^e}{\partial z} \\ 0 & \frac{\partial N_1^e}{\partial z} & \frac{\partial N_1^e}{\partial y} & \cdots & 0 & \frac{\partial N_{ne}^e}{\partial z} & \frac{\partial N_{ne}^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & 0 & \frac{\partial N_1^e}{\partial x} & \cdots & \frac{\partial N_{ne}^e}{\partial z} & 0 & \frac{\partial N_{ne}^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & 0 & \cdots & \frac{\partial N_{ne}^e}{\partial y} & \frac{\partial N_{ne}^e}{\partial x} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1^e & \mathbf{B}_2^e & \cdots & \mathbf{B}_{n^e}^e \end{bmatrix} \tag{3.32}$$

In order to illustrate this section, which may seem a little bit abstract to the reader, let's return to the **example** presented in figure 3.1. Imagine a 1D problem, where the domain has been split into smaller straight lines (elements). Each element has at least two nodes where the displacements have to be computed. It is not the only possible case: one can compute the displacements at more than two points by increasing the number of nodes of the element. In 1D, an element with two nodes is called linear; if it has three nodes, quadratic; if four, cubic; and so on. These names refer to the **order** of the polynomial that has to be used as shape function in each case.

The key is that no matter the length of the element h^e in the physical domain. When mapped to the parent domain, the element boundaries go from $\xi = -1$ to $\xi = +1$. Is in this parent domain where the shape functions $\mathbf{N}^e$ are defined: as the element length is normalised, the shape functions can be the same for **all** elements. Recall that in the FEM, a shape function N_a^e is restricted to have a value of **1** at node **a**, and **0** at every other node. Notice that in order to accomplish the latter, there will be as many shape functions as nodes. With all these properties in mind, the following figures represent shape functions of different order for a 1D element defined in the parent domain:



(a)

(b)

(c)

Figure 3.2: 1D shape functions for linear (a), quadratic (b) and cubic (c) isoparametric elements.

The same reasoning that has been followed for a 1D element can be developed for 2D and 3D elements. In those cases, many different types of elements can be defined, depending on their geometric properties and number of nodes. The shape functions will then have two or three spatial variables, and are to be defined for each type of element. For instance, in order to complete the previous example, let's consider the case of a linear triangular element:

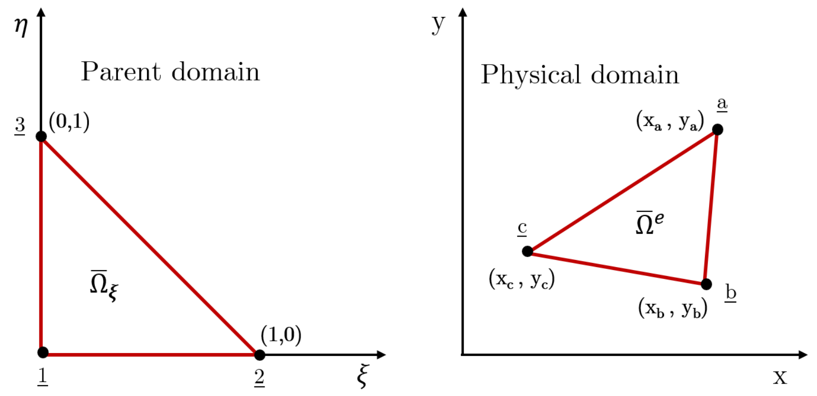


Figure 3.3: Representation of a triangular linear element in both parent and physical domains

In the parent domain, the nodes are always listed in **counter-clockwise** direction. As can be seen, no matter the shape of the triangle in the physical domain, once it is mapped into the parent space all the elements are geometrically identical. The matrix of shape functions of the above element and its gradient in the parent domain are, then:

$$\mathcal{N}^e = \left[\; 1{-}\xi{-}\eta \quad \xi \quad \eta \;\right], \;\; \tilde{\boldsymbol{B}}^e_\xi = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.33}$$

3.3.1. The Gauss Quadrature integration method

Although the domain has been divided into smaller elements, the integrals presented earlier are still difficult to compute, as those elements have a wide variety of **different boundaries**. If the geometry of the element is sort of complex (nothing unusual in 3D), the integrals would have to be evaluated along those boundaries, which also change from one element to another. The solution to this problem has been presented in the previous section: if the element is mapped into the parent domain, its geometric properties are kind of normalised. The integrals can then be evaluated in the parent domain, and later on be transformed into global coordinates by using the Jacobian matrix (recall equation 3.30).

Now the question lies in which method use to evaluate those integrals. For sure, the analytic solution is discarded as it would take too much time. From among all the numeric integration methods, one of the most widespread is Gauss Quadrature. It is a fast and simple method, that give great approximations with few operations. Basically, it reduces the integral to a **summation** of function values at specific points (Gauss points), multiplied by a certain factor called weight. The great advantage of this method is that as all the elements of the same type are mapped equally in the parent domain, the values of the weights and gauss points will be the same for all elements.

Imagine that the integral to evaluate I is of some function $f : \overline{\mho}^e \to \mathbb{R}$ over the physical domain. First, the integral has to be mapped into the parent domain through a change of variable, accomplished by using the Jacobian matrix of the transformation $\boldsymbol{x} = \boldsymbol{x}(\xi)$:

$$I = \int_{\mho^e} f \, d\mho = \int_{\mho_\xi} \mathtt{J}^e(\xi) f(\xi) \, d\mho_\xi \tag{3.34}$$

Where $\mathtt{J}$ stands for the determinant of the Jacobian. In 3D, $\mathrm{d}\Omega_\xi = \mathrm{d}\xi \; \mathrm{d}\eta \; \mathrm{d}\zeta$. As Gaussian rules for integrals in several dimension are constructed by using 1D Gaussian rules on each coordinate separately, hereinafter only the one-dimensional Gauss Quadrature will be considered. The goal of the method is to find a set of Gauss points ξ_g and weights w_g such that:

$$I = \int_{\mho_\xi} \mathtt{J}^e(\xi) f(\xi) \, d\mho_\xi \approx \sum_{g=1}^{m} w_g \mathtt{J}^e(\xi_g) f\left(\xi_g\right) \tag{3.35}$$

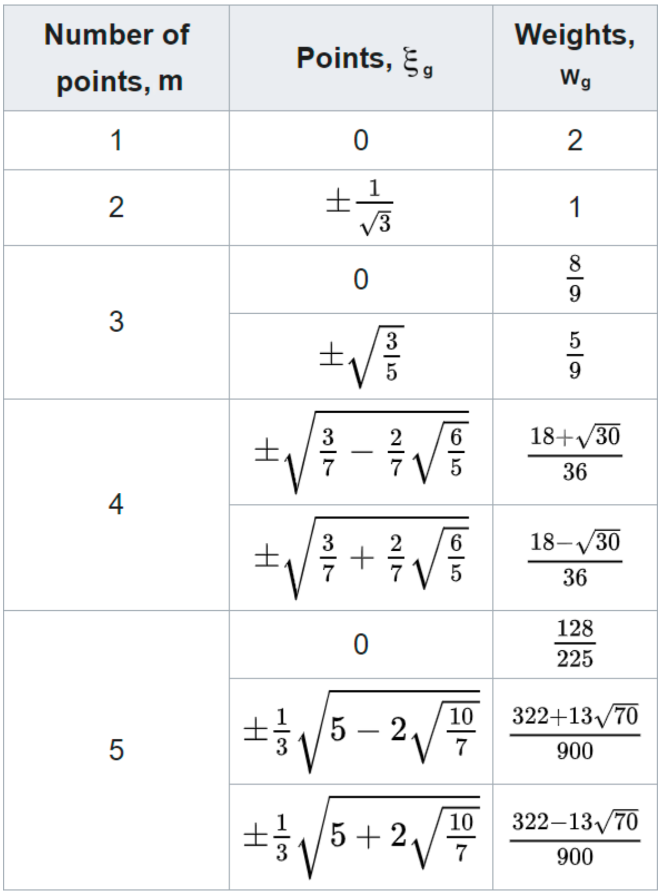
The above expression give exact results only if $\mathtt{J}^e(\xi) f(\xi)$ is a **polynomial**. If not, only approximate results are obtained, whose accuracy improves as more Gauss points are used. Supposing that the integrand $q(\xi) = \mathbf{J}^e(\xi) f(\xi)$ is a polynomial of order p:

$$q(\xi) = \sum_{i=0}^{p} \alpha_i \xi^i \;\; \Rightarrow \;\; I = \sum_{i=0}^{p} \alpha_i \int_{\mho_\xi} \xi^i \, d\mho_\xi = \sum_{i=0}^{p} \alpha_i \left(\sum_{g=1}^{m} w_g \xi_g^i \right) \tag{3.36}$$

The values of the Gauss points and weights are obtained by **exactly** evaluating each monomial ξ^i, with $i = 1, 2 \ldots p$:

$$\sum_{g=1}^{m} w_g \xi_g^i = \left.\frac{\xi^{i+1}}{i+1}\right|_{\Omega_\xi} \xrightarrow{\text{if } \Omega_\xi = [-1,+1]} = \begin{cases} 0 & \text{if} \quad i = 1,3,5\ldots \\ \frac{2}{i+1} & \text{if} \quad i = 0,2,4,6\ldots \end{cases} \tag{3.37}$$

The previous expressions lead to a **linear system** of $p+1$ equations with $2m$ unknowns. If q results to be a polynomial, the exact integration requires from $m \geq \frac{p+1}{2}$ points. Gauss points and weights will be different depending on the number of points m used. The values of the Gauss points and weights up to 5 points are presented in the following table:



Number of points, m	Points, ξ_g	Weights, w_g
1	0	2
2	$\pm\frac{1}{\sqrt{3}}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18-\sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$

Table 3.1: Gauss points positions and weights [9]

4

FEM programming

Now that the elastodynamic model for rotating structures has been formulated in terms of the FEM, it is time to tackle its programming. This section covers the different processes involved in the simulation and how the MATLAB solver works. In this chapter, only the static case is to be addressed, as the methods used to solve the dynamic problem will be reviewed later in the report.

4.1. Preprocessing

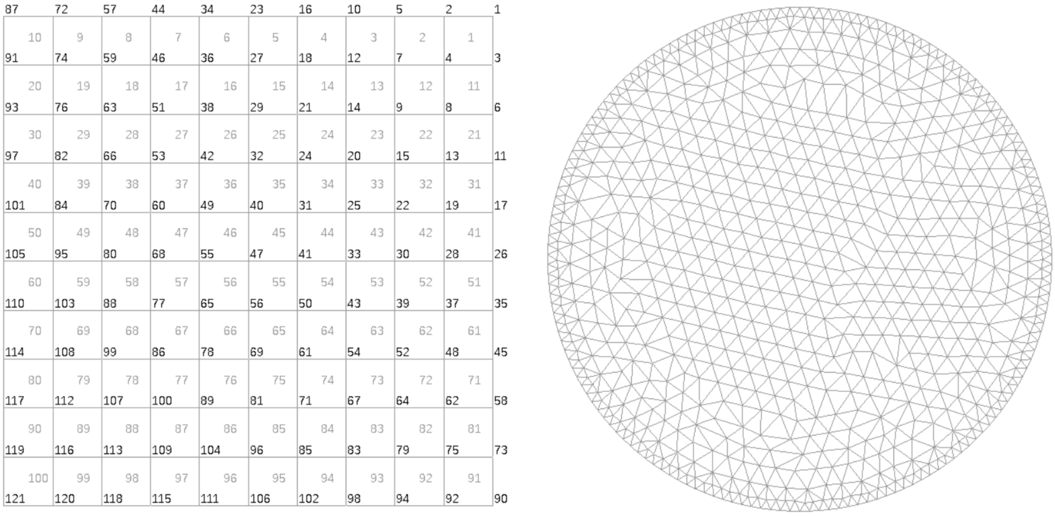
All the finite element simulations, carried out or not by commercial codes, start with the model definition, a critical phase widely known as preprocessing. This first step begins with the definition of the **geometric domain** of the problem: in this case, the structure. Nowadays, the most widespread method to define the geometry is by using computer-aided design (CAD) software. Examples of commercial CAD programs are *CATIA*, *Rhino* and *SolidWorks*.

Once the geometry is defined, it is exported to what is known as mesh generation software. The final goal of the latter is to create a **mesh**: a subdivision of the geometry into discrete cells known as elements. As it has been commented in the previous chapter, many different types of elements exist, so its type, length, area and other properties are to be defined. Notice that meshing is not a trivial process as geometries can be quite intricate. In order to accurately capture the input domain geometry, the mesh generation is often aided by computer algorithms. In this step, the domain boundary is also to be defined. Once the mesh has been generated, the software computes what is known as **connectivity matrix**, which contains all the information about the mesh and defines the relations between elements and nodes (see for instance the numbers printed in figure 4.1a).

The connectivity matrix **CN** $\in \mathbb{R}^{n_{el} \times n^e}$, also known as element connectivity array, stores all the element definitions. It is an array of integers numbers where the $(\mathbf{e},\mathbf{a})$ cell contains the global number of the $\mathbf{a}$—th node of element **e**. It is an essential matrix as it stores all the information regarding the mesh, and it is crucial when transforming from global to local coordinates.

Quite often, commercial programs allow both designing and meshing. This is in fact the case of **GiD**, the pre and postprocessing tool used during this project. However, its designing tools are quite basic, and so it admits the importation of models from other CAD softwares. Nonetheless, those tools are quite useful when editing an imported geometry in order to simplify the meshing procedure. Figure 4.1 illustrates the meshing of a 2D figure with GiD, using triangular and quadrilateral elements. In some cases, the mesh can be very simple (for instance a grid in figure 4.1a), whereas other geometries require from more complex meshes.

The next step is to define, for each element, material properties (density, elastic coefficients, heat conductivity, etc.) and loadings (volumetric forces, internal heat sources, etc.). The final part of the preprocessing phase consists in defining boundary conditions, which in our case is to prescribe displacements and tractions along the boundary. Commercial softwares usually include a wide library of material properties and loading scenarios to carry out these last steps.



(a)

(b)

Figure 4.1: Mesh representation using quadrilateral (a) and triangular (b) elements. In the case of the circle, the mesh has been generated using in-built algorithms.

As GiD does not implement this capability, this last part of the preprocessing is carried out with MATLAB. Once the mesh is created, GiD exports a raw file (*.msh* extension) including the connectivity matrix, the nodal coordinates, the element typology and boundary information. Those are later recovered and imported to MATLAB. Material properties are manually assigned to each element, as well as the nodal values of body forces and prescribed boundary displacements and tractions.

4.2. Solver

Once the formulation of the problem is completed and all the information regarding structure's geometry, material properties, loads and boundary conditions is gathered in a MATLAB file, it is time to solve the elastostatic equations. In fact, the equations to be solved correspond to the approximated weak form of the problem stated in terms of the FEM (see equation 3.21). In this chapter, only the **static form** of the problem is to be tackled — that is, $\ddot{\mathbf{d}} = \dot{\mathbf{d}} = 0$:

$$\mathbf{d}_\text{l} = \mathbf{K}_\text{ll}^{-1} \left(\mathbf{F}_\text{l} - \mathbf{K}_\text{lr}\, \overline{\mathbf{u}} \right) \tag{4.1}$$

It is a first step in the understanding of the whole phenomena, and the developed code is going to be re-used later on when studying the dynamic case. The code is developed in MATLAB, with its main working algorithm schematised in the following figure:

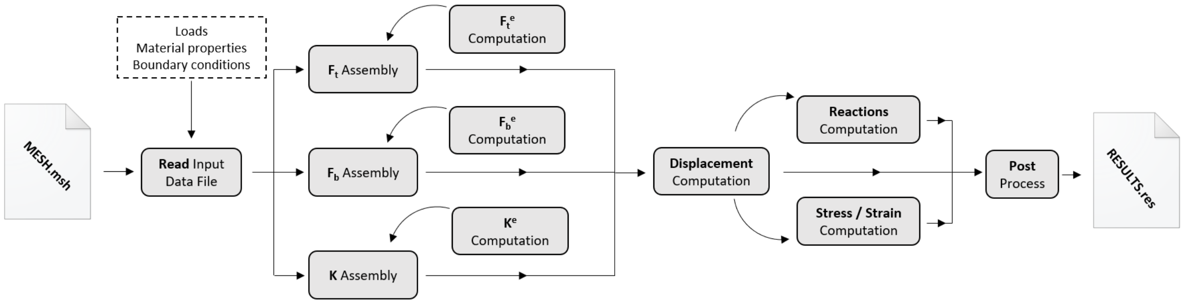


Figure 4.2: MATLAB main code scheme for elastostatic problems

As it can be suspected from the previous figure, the main task of the solver is to compute the matrix **K** and vector **F** from equation 3.13. Using the local formulation presented in the previous chapter, the computation of the latter can be split into a simpler problem, regarding only **elemental** matrices $\mathbf{K}^e$ and vectors $\mathbf{F}^e$. Once they are obtained for all the elements of the domain, the elemental matrices are joined together into a global stiffness matrix **K** and force vector **F** through a process called assembly. When **K** and **F** are known, equation 4.1 results in a linear system of algebraic equations, easily handled with native MATLAB functions. A further explanation regarding elemental matrix computation and assembly can be found in Annex B.

Once the displacement field is known, the final step of the solver consists in finding the reactions and the strain field. Computation of **elastic reactions** is straightforward, as they are obtained from equation 3.21. On the other hand, the **strain field** is to be computed in the local frame, element by element. Strains are defined as the spatial derivative of displacements (see equation A.7), which can be written in terms of the FEM as:

$$\boldsymbol{\varepsilon} = \nabla^s \mathbf{u} \;\Rightarrow\; \boldsymbol{\varepsilon}^h = \nabla^s \mathbf{u}^h = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \mathbf{B}^e \mathbf{d}^e \;\Rightarrow\; \boldsymbol{\sigma}^h = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \mathbf{C}\,\mathbf{B}^e \mathbf{d}^e \tag{4.2}$$

Where **A** represents the assembly operator. As $\mathbf{B}^e$ is approximated through Gauss Quadrature, strains and stresses are only computed at the Gauss points ξ_g.

4.2.1. Efficient assembly

In the FEM, computation of global matrices is never done in a single step. Instead, elemental matrices are computed element by element, and then joined together through a process called **assembly**, constructing this way the global matrix. The intuitive method to perform the assembly is quite simple to program: elemental matrices are computed one by one, and then allocated into a bigger, global matrix using the connectivity array. The connectivity array **CN** links the local numeration of the degrees of freedom with the global one, specifying the row and column of the global matrix where each component of the elemental matrix has to be placed. For a one-dimensional case, the classical assembly of a matrix **M** can be programmed as follows:

```
for e = 1 : n_el %Loop over all the elements
   M_e = Compute_Me(. . .); %Function that computes the elemental matrix
   for a = 1 : n_e %Loop over the nodes of the element e
       for b = 1 : n_e
          A = CN(e,a); B = CN(e,b); %Transformation from local to global
          M_glo(A,B) = M_glo(A,B)+M_e(a,b); %Allocate M_e into the global matrix
       end
   end
end
```

The previous algorithm has two main drawbacks. The first one is related with the **memory** large matrices take up. Each cell of a MATLAB matrix is defined as *double*, a type of variable that uses 8 bytes of memory. Using the traditional approach, a 3D system with 30000 nodes would involve matrices with a size bigger than 7.5 gigabytes, which is clearly infeasible. But if one takes closer look to the matrices involved in the FEM, will discover that most of the cells are filled with zeros. Matrices in which most of the elements are zero are known as *sparse* matrices.

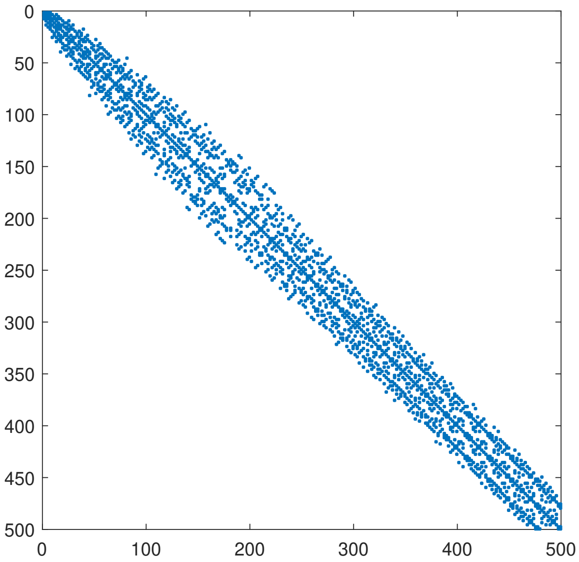


Figure 4.3: Representation of a sparse matrix. Blue squares are non-zero values.

For sure, storing the zeros of a sparse matrix as double variables is a waste of memory. That is why most of the programming languages have in-built algorithms to store sparse matrices in more efficient ways. The key is to store only the non-zero values and their position in the matrix. Using this approach, systems with a large number of degrees of freedom can be tackled without memory problems.

The second disadvantage of the classical assembly method is related with its **computational cost**. A closer look to the presented algorithm reveals three nested loops. A local matrix has to be computed for each element, and then placed into the global matrix cell by cell. In MATLAB, this iteration is rather slow. For the static case, where matrices are computed once, it is not a big deal. But when dealing with the dynamic case, global matrices are to be computed repeatedly, increasing considerably the running time of the solver. This problem is solved by using a **vectorised** version of the assembly algorithm.

The key of the vectorised algorithm is that computes and assembles all the elemental matrices at once, avoiding iterations [10]. It is more difficult to program than the classical algorithm, but its advantages are considerable. For instance, figure 4.4 reveals an outstanding decrease of the computation time for 2D systems. As it can be spotted, although for few degrees of freedom the classical algorithm is quicker, the vectorised algorithm proves to be the most efficient solution as the size of the system grows.

The vectorised assembly algorithm has been adapted for the rotating case from a set of codes given by prof. *Hernández J.*, which were used in the context of composite materials. An exemplification of the vectorised assembly for a simple mesh can be found in Annex C.

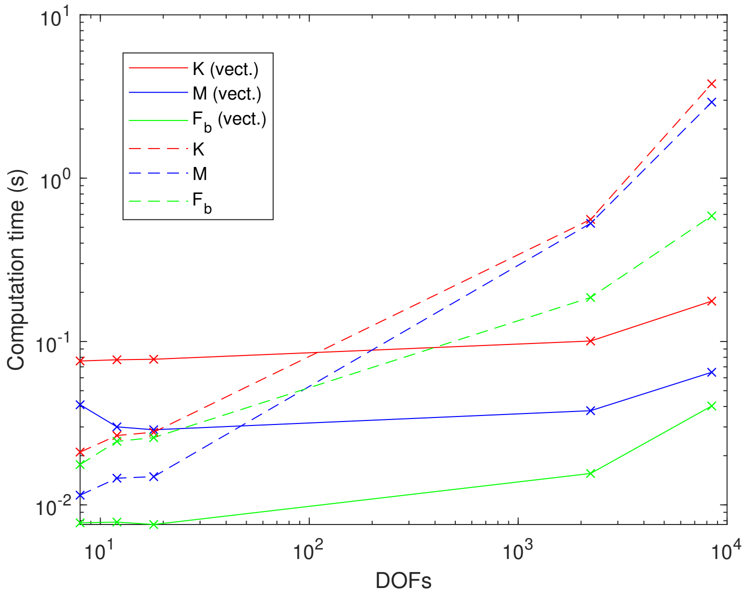


Figure 4.4: Comparison of the computation time of **F**, **M** and **K** using the classical and vectorised assembly (logarithmic scale).

4.2.2. Approximated one-dimensional stationary solution

To conclude the *solver* section, let's exemplify the explained concepts approximating the solution of the system presented in 2.26. It is the simplest case to tackle: a 1D blade rotating at constant speed with no vibration at all. It can be interpreted as a static elastic problem in the rotational frame, where deformations can only occur in the axial direction ς. The goal is to find the **approximated solution** using the FEM, and assess its validity by comparing it with the analytic result obtained in 2.5.2. Recall the differential equation to solve:

$$\frac{d^2\mathbf{u}_\varsigma}{d\varsigma^2} = -\frac{1}{\mathrm{E}}\,\rho\,\Omega^2\,(\varsigma + \mathbf{u}_\varsigma) \tag{4.3}$$

Using the FEM formulation, an approximated weak form of the equation can be written:

$$-\int_\mho \nabla^s\mathbf{v}^{h^T}\;\mathrm{E}\;\nabla^s\mathbf{u}_\varsigma^h\,d\varsigma + \int_\mho \mathbf{v}^{h^T}\;\rho\,\Omega^2\,\mathbf{u}_\varsigma^h\;d\varsigma + \int_\mho \mathbf{v}^{h^T}\;\rho\,\Omega^2\;\varsigma d\varsigma = 0 \tag{4.4}$$

By replacing the test and trial function with its approximations in terms of the shape functions, the above equation yields to:

$$\overbrace{\left(\underbrace{\int_\mho \mathbf{B}^T\;\mathrm{E}\;\mathbf{B}\;d\varsigma}_{\mathbf{K}_\mathrm{static}} - \underbrace{\int_\mho \mathbf{N}^T\;\rho\,\Omega^2\,\mathbf{N}\;d\varsigma}_{\mathbf{K}_\mathrm{rot}}\right)}^{\mathbf{K}}\mathbf{d} = \overbrace{\int_\mho \mathbf{N}^T\;\rho\,\Omega^2\;\varsigma\;d\varsigma}^{\mathbf{F}=\mathbf{F}_\mathrm{rot}} \tag{4.5}$$

Taking into account that displacements are prescribed along the set **r** and unknown in **l**, the above equation can be split into block matrices:

$$\mathbf{d}_\mathrm{l} = \mathbf{K}_\mathrm{ll}^{-1}\left(\mathbf{F}_\mathrm{l} - \mathbf{K}_\mathrm{lr}\,\bar{\mathbf{u}}_\varsigma\right) \tag{4.6}$$

Previous to the solving of the above system of algebraic linear equations, global matrices **K** and **F** are to be known. As it has been discussed, the best choice is to decompose the latter into elemental matrices:

$$\mathbf{K} = \underset{e=1}{\overset{n_{el}}{\mathbf{A}}} \mathbf{K}^e \ , \quad \mathbf{F} = \underset{e=1}{\overset{n_{el}}{\mathbf{A}}} \mathbf{F}^e \tag{4.7}$$

Elemental matrices are to be calculated using the approach reviewed in Annex B. The first step is to define the type of element in the parent domain. For simplicity, let's assume **linear elements**. Recall the shape functions in the parent domain for a linear 1D element from figure 3.2(a):

$$\mathbf{N}^e_\xi = \frac{1}{2}\left[(1-\xi) \;\; (1+\xi)\right] \ , \quad \mathbf{B}^e_\xi = \frac{\mathrm{d}\mathbf{N}^e_\xi}{\mathrm{d}\varsigma} = \frac{\mathrm{d}\mathbf{N}^e_\xi}{\mathrm{d}\xi}\,\frac{\mathrm{d}\xi}{\mathrm{d}\varsigma} \tag{4.8}$$

To compute $\mathbf{B}^e_\xi$, the Jacobian of the transformation $\frac{\mathrm{d}\xi}{\mathrm{d}\varsigma}$ is to be found:

$$\frac{\mathrm{d}\xi}{\mathrm{d}\varsigma} = \left(\frac{\mathrm{d}\mathbf{N}^e}{\mathrm{d}\xi}\boldsymbol{\varsigma}^e\right)^{-1} = 2\left(-\varsigma^e_1 + \varsigma^e_2\right)^{-1} = \frac{2}{h^e} \tag{4.9}$$

Where h^e represents the length of the element. Then:

$$\mathbf{B}^e_\xi = \frac{1}{h^e}\left[-1 \quad +1\right] \tag{4.10}$$

With all, the computation of the elemental stiffness matrix is reduced to:

$$\mathbf{K}^e_{\text{static}} = \int_{\mho} \mathbf{B}^{e^T}\,\mathrm{E}\,\mathbf{B}^e\; d\varsigma = \int_{-1}^{1} \frac{1}{h^e}\left[-1 \quad +1\right]^T \mathrm{E}\,\frac{1}{h^e}\left[-1 \quad +1\right]\,\frac{h^e}{2} d\xi = \frac{\mathrm{E}}{h^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\begin{aligned}\mathbf{K}^e_{\text{rot}} &= -\int_{\mho} \mathbf{N}^{e^T}\,\rho\,\Omega^2\,\mathbf{N}^e\; d\varsigma = -\,\rho\,\Omega^2\int_{-1}^{1}\frac{1}{2}\left[(1-\xi) \;\; (1+\xi)\right]^T\frac{1}{2}\left[(1-\xi) \;\; (1+\xi)\right]\,\frac{h^e}{2}d\xi = \\ &= -\,\rho\,\Omega^2\frac{h^e}{8}\int_{-1}^{1}\begin{bmatrix}(1-\xi)^2 & (1-\xi)\,(1+\xi) \\ (1-\xi)\,(1+\xi) & (1+\xi)^2\end{bmatrix} d\xi = -\rho\,\Omega^2\frac{h^e}{6}\begin{bmatrix}2 & 1 \\ 1 & 2\end{bmatrix}\end{aligned} \tag{4.11}$$

Recall that for the calculation of the elemental stiffness matrix, integrals are evaluated exactly. This is not the case of the force vector, which is approximated trough Gauss Quadrature. Replacing $\rho\,\Omega^2\,\varsigma$ by **f**:

$$\begin{aligned}\mathbf{F}^e &= \int_{\mho}\mathbf{N}^{e^T}\,\mathbf{f}\,d\varsigma = \int_{\mho}\mathbf{N}^{e^T}\left(\mathbf{N}^e\mathbf{f}^e\right)\,d\varsigma = \int_{-1}^{1}\mathbf{N}^{e^T}\mathbf{N}^e\rho^e\,\Omega^2\,\varsigma^e\frac{h^e}{2}d\xi = \\ &= \frac{h^e}{8}\sum_{g=1}^{m} w_g\rho^e\,\Omega^2\begin{bmatrix}(1-\xi_g)^2 & (1-\xi_g)\,(1+\xi_g) \\ (1-\xi_g)\,(1+\xi_g) & (1+\xi_g)^2\end{bmatrix}\begin{bmatrix}\varsigma^e_1 \\ \varsigma^e_2\end{bmatrix}\end{aligned} \tag{4.12}$$

As $\mathbf{N}^e$ is linear in ξ and **f** is also linear in ς, three Gauss points ($m = 3$) will suffice to evaluate the above integral exactly. The values of Gauss points and weights can be found in table 3.1. Once the elemental stiffness matrix and force vector are known, they are computed for each element of the domain and assembled into global matrix **K** and vector **F**. The last step before finding displacement field is to solve the system of equations stated in 4.6.

In order to **compare** the results with the exact solution, let's consider the same conditions as in the analytic case. The structure of study will be a blade with constant cross-section and 4 m radius, rotating at 60 rad/s. For an aluminium bar ($E = 70\ GPa,\ \rho = 2700\ kg/m^3$), the results are displayed in figure 4.5 and compared with the exact solution. Recall that computation of stresses once nodal displacements are known is achieved using expression 4.2.

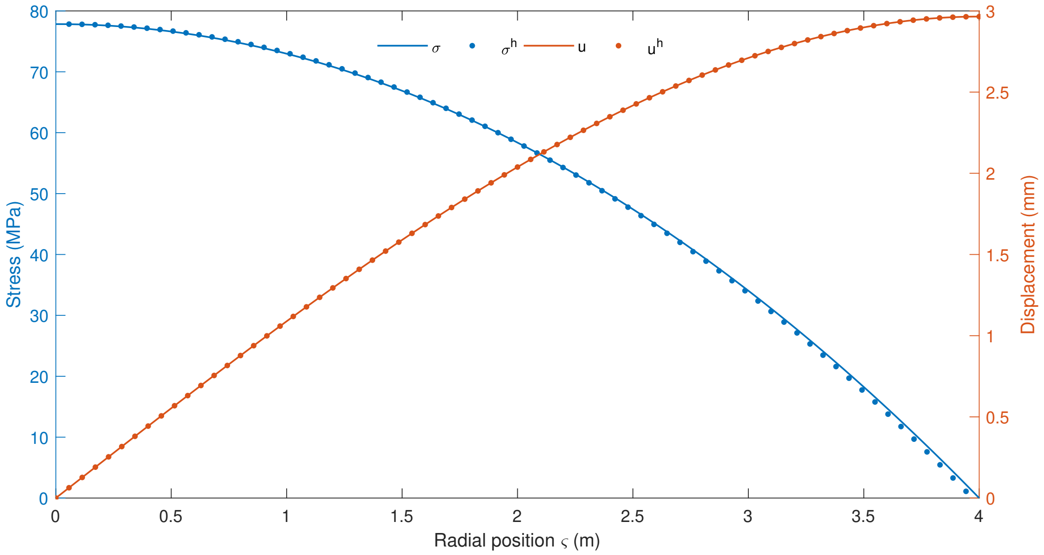


Figure 4.5: Approximated solution of equation 2.26 using 70 elements, compared with the exact result.

In the FEM, another input parameter to define is the number of elements into which the domain is divided. Theoretically, the approximated solution **converges** into the exact one as the number of elements increases. In order to assess this convergence, let's define the error associated to the approximated solution **e** using the **L2 norm** of the distance between $\mathbf{u}_\varsigma$ and $\mathbf{u}_\varsigma^h$:

$$\|\mathbf{e}\|_{L_2} = \left\|\mathbf{u}_\varsigma - \mathbf{u}_\varsigma^h\right\|_{L_2} = \left(\int_{\mho}\left(\mathbf{u}_\varsigma - \mathbf{u}_\varsigma^h\right)^2 d\varsigma\right)^{\frac{1}{2}} \tag{4.13}$$

In a similar way, one can define the error **e'** associated to the derivative of **u**, which in this case is linked with stresses. In order to evaluate those integrals, the domain is again split into elements where Gauss Quadrature is applied. As quite often the exact solution is not a polynomial, high order Gauss rules are needed. In general, the logarithm of the error varies linearly with the element size, and its slope depends on the order of the element:

$$\|\mathbf{e}\|_{L_2} = Ch^{\alpha} \tag{4.14}$$

Where α is known as the rate of convergence of the element, while C is an arbitrary constant and h the element length (assumed constant). The parameter α is equal to $p+1$, where p is the order of the used shape function. For the derivative of the error, the slope of convergence is one order lower. Recall that if obtaining the analytic exact solution were impossible, the computation of the error in the way that has been developed would not be practicable. In that case, in order to check the convergence of the method, **relative error** should be computed. That is, to compare the error of the function obtained using n_{el} elements, with a previous function obtained with $n_{el} - k$ elements.

4.3. Postprocessing

The last step of every FEM simulation is analysis and evaluation of the solution results, often referred as postprocessing. Once the solution of the problem is computed, it is exported to a postprocess program that interprets, prints and plots the results. Among many other operations, these programs are used to **draw** the deformed structural shape, produce colour-coded plots (representing displacements, strains and stresses) and **animate** the dynamic behaviour of the system, if that were the case. Equilibrium check (assessing if reactions counteract external forces, for example) is also part of the postprocess stage.

In the previous example, results were graphed in a simple plot. This approach is unattainable for multidimensional problems with non-trivial geometry, and thus external programs are to be used. In this project, the postprocessing is carried out by **GiD**, the same program used for the mesh definition. To illustrate **GiD** capabilities, let's consider a fixed-free cantilever beam under the effect of a distributed load $w = -20\ KN/m$ along its upper side. The beam is shaped as a long rectangle $(0,2 \times 2\ m)$ and meshed using 1000 linear quadrilateral elements. The following figure represents its **deformed configuration**, and stresses along the beam are represented using colours:

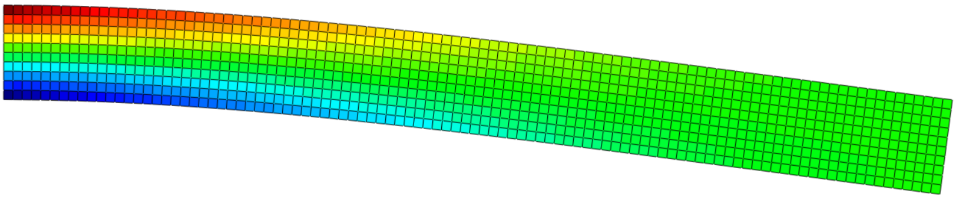


Figure 4.6: Deformed shape (×200) and stress (σ_x) colour map of a 2D beam under the effect of a distributed load on its upper side.

The colour legend has not been displayed as it is not from interest in this example. Remember that although displacement is continuous along the domain, its derivatives (stress and strains) are only computed at Gauss points. Modern postprocessing programs have in-built tools that blur the latter in order to hide discontinuities. The engineer has to be aware that misuse of blurring tools can induce notable errors.

To both asses the accuracy and validate the 2D solver, FEM results can be contrasted with the **analytic solution** obtained from beam theory. In particular, vertical displacement along the mean line of the beam is the variable that is being compared in figure 4.7. Analytic results are those of a beam with squared cross section, whose deflection can be expressed as:

$$y = \frac{-wx^2}{24EI}\left(x^2 - 4Lx + 6L^2\right) \tag{4.15}$$

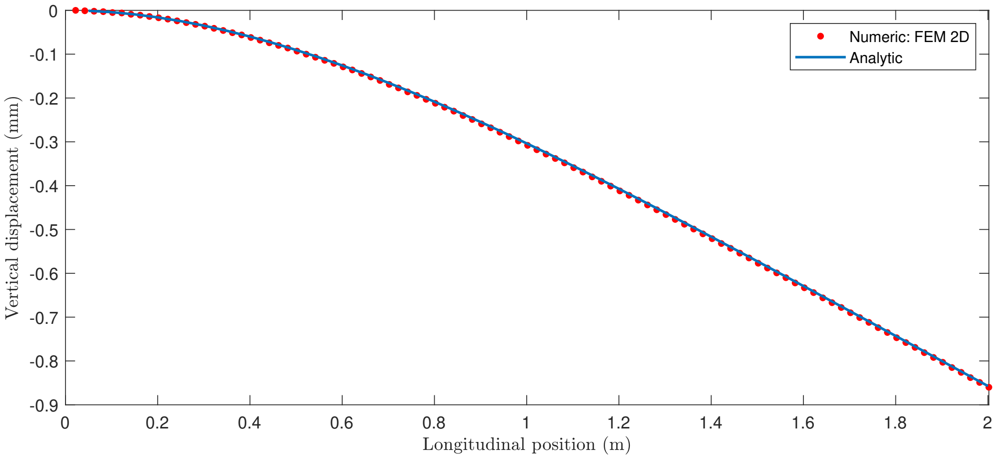


Figure 4.7: Comparison of the approximated and theoretical vertical displacement of a beam under the effect of a distributed load.

5

Dynamic response

So far, the problem regarding static deformations in the noninertial frame is fully settled. However, although it can be used to determine initial conditions, the developed model is unable to account for **time-dependant variables** such as velocity or acceleration in the rotating frame. In other words: it cannot capture the phenomena of vibration. In order to study the dynamic response of the system, new approaches are to be considered. Hence, this chapter will review two main methodologies to tackle time-dependant problems: modal decomposition analysis and numerical time integration. A brief description regarding the singular value decomposition (SVD) will be presented at the end of this chapter, presenting a promising method to handle nonlinear dynamic systems.

5.1. Modal decomposition analysis

Modal analysis is one of the most widespread techniques to tackle structural vibration problems, for both discrete and continuous systems. The key of this method is that it studies the problem in the **frequency domain**, rather than in the temporal one. To start with, recall the equation describing a rotating structure in the noninertial frame of reference:

$$\mathbf{M_{ll}}\,\ddot{\mathbf{d}}_{\mathbf{l}} + \mathbf{D_{ll}}\,\dot{\mathbf{d}}_{\mathbf{l}} + \mathbf{K_{ll}}\,\mathbf{d}_{\mathbf{l}} = \mathbf{F_l} - (\,\mathbf{M_{lr}}\,\overline{\ddot{\mathbf{u}}} + \mathbf{D_{lr}}\,\overline{\dot{\mathbf{u}}} + \mathbf{K_{lr}}\,\overline{\mathbf{u}}) \tag{5.1}$$

Where **K**, **D** and **F** are the sum of two contributions: a first one typical of static conditions, and a particular second one accounting for the effect of rotation. For the sake of notational simplicity, on now on let's assume the following redefinitions:

$$\begin{gathered}\mathbf{M_{ll}} \rightarrow \mathbf{M}\ , \quad \mathbf{K_{ll}} \rightarrow \mathbf{K}\ , \quad \mathbf{D_{ll}} \rightarrow \mathbf{D}\ , \quad \mathbf{F_l} - (\,\mathbf{M_{lr}}\,\overline{\ddot{\mathbf{u}}} + \mathbf{D_{lr}}\,\overline{\dot{\mathbf{u}}} + \mathbf{K_{lr}}\,\overline{\mathbf{u}}) \rightarrow \mathbf{F} \\ \mathbf{d_l} \rightarrow \mathbf{d}\ , \quad \mathbf{d}_{\mathbf{l}}^{0} \rightarrow \mathbf{d}^{0}\ , \quad \dot{\mathbf{d}}_{\mathbf{l}}^{0} \rightarrow \dot{\mathbf{d}}^{0}\end{gathered} \tag{5.2}$$

The advantage of the presented reformulation is that allows equation 5.1 to be expressed as:

$$\mathbf{M}\,\ddot{\mathbf{d}} + \mathbf{D}\,\dot{\mathbf{d}} + \mathbf{K}\,\mathbf{d} = \mathbf{F} \tag{5.3}$$

With $\mathbf{d} : [0,\mathrm{T}] \rightarrow \mathbb{R}^n$ and subject to initial conditions $\mathbf{d}_{(0)} = \mathbf{d}^0$ and $\dot{\mathbf{d}}_{(0)} = \dot{\mathbf{d}}^0$. The previous expression can be thought as a second order ordinary linear differential equation. From calculus, it is known that the solution of this kind of ODE is the sum of a particular $\mathbf{d^p}$ and homogeneous $\mathbf{d^z}$ solution:

$$\mathbf{d} = \mathbf{d^z} + \mathbf{d^p} \tag{5.4}$$

The homogeneous solution of the system is found by making **F** = 0, whereas the particular solution is to be computed for the specific case of study. One of the main drawbacks of the modal analysis is that, as it works in the frequency domain, forcing functions need to be **harmonic**: $\mathbf{F} = \overline{\mathrm{F}}\,\sin(\mathrm{w}\ \mathrm{t})$. This is not the case of the system of study, where forces due to rotation are either constant or increasing in time, and external forces can adopt any arbitrary shape. One way of tackling this problem is to decompose **F** into the sum of harmonic functions through the **Fourier transform**. As the system of study is linear, its response can be expressed as the sum of the responses due to each of the obtained harmonic forcing functions.

If **F** is difficult to pose in terms of harmonic functions, another approach consists in splitting **F** into the sum of *n* **impulses**. The response of the system to an impulse is identical to the free response with certain initial conditions [11]. Again, the system is studied under the effect of each of those individual impulses, and taking advantage of the linearity of the system, the solution is expressed as the sum of the latter individual responses. As modal study of structures under a generic forcing function **F** can easily become complicated, the present section will only cover the homogeneous solution of equation 5.3.

5.1.1. Undamped free vibrations

In order to introduce the basics of the modal analysis, let's consider the simplest case to handle: a system where both **F** and **D** are zero, so damping and external forces play no role whatsoever. With this assumption, equation 5.3 reduces to:

$$\mathbf{M}\,\ddot{\mathbf{d}} + \mathbf{K}\,\mathbf{d} = 0 \tag{5.5}$$

The modal analysis starts supposing a solution with the following form:

$$\mathbf{d} = \boldsymbol{\Phi}\ \mathbf{q}_{(t)} = \boldsymbol{\Phi}\ (\mathbf{A}\cos(\boldsymbol{\omega}\ \mathrm{t}) + \mathbf{B}\sin(\boldsymbol{\omega}\ \mathrm{t}))\quad ,\quad \boldsymbol{\Phi} \in \mathbb{R}^n \tag{5.6}$$

Where $\mathbf{q}_{(t)}$ is a known solution for homogeneous second order differential equations. Since $\ddot{\mathbf{q}} = -\boldsymbol{\omega}^2\ \mathbf{q}$, the above equation can be written as:

$$\left(\mathbf{K} - \boldsymbol{\omega}^2\,\mathbf{M}\right)\boldsymbol{\Phi}\ \mathrm{q}_{(t)} = 0 \quad \mapsto \quad \left(\mathbf{K} - \boldsymbol{\omega}^2\,\mathbf{M}\right)\boldsymbol{\Phi} = 0 \tag{5.7}$$

The preceding equation is known as the *eigenvalue problem*. The trivial solution **Φ** = 0 is not from interest, whereas the non-trivial solution is given by a set of eigenvalues $\boldsymbol{\lambda} = \boldsymbol{\omega}^2$, where $\boldsymbol{\omega}$ represents the diagonal matrix of natural or resonant vibration frequencies. For a system with **n** degrees of freedom, **n** eigenvalues exist that satisfy the above expression. Being j an index that goes from 1 to **n**, the eigenvalue problem can be interpreted as a system of equations of the form

$$\left(\mathbf{K} - \lambda_j\,\mathbf{M}\right)\boldsymbol{\Phi}_j = 0 \tag{5.8}$$

Where $\boldsymbol{\Phi}_j$ is referred as the *eigenvector* associated to the eigenvalue λ_j. Notice that eigenvectors do never define absolute values of the different independent coordinates of the system, but inform about the phase and amplitude **relations** between degrees of freedom. In structural problems, eigenvectors are called natural modes or mode shapes, and can be though as possible deformed configurations oscillating at a given frequency stated by its corresponding eigenvalue. In matrix form, eigenvalues and eigenvectors are expressed as:

$$\boldsymbol{\lambda} = \begin{pmatrix} {\omega_1}^2 & 0 & \cdots & 0 \\ 0 & {\omega_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & {\omega_n}^2 \end{pmatrix} \quad , \quad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \cdots & \boldsymbol{\Phi}_n \end{bmatrix} = \begin{pmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1n} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{n1} & \Phi_{n2} & \cdots & \Phi_{nn} \end{pmatrix} \tag{5.9}$$

The main advantage of this method is that natural frequencies and mode shapes are relatively easy to obtain. MATLAB, for instance, has in-built functions that compute a set of $\mathbf{m} \leq \mathbf{n}$ eigenvectors with the smallest associated frequency. The key of this approach is that there is no need to compute all the **n** eigenvalues and eigenvectors, and instead the solution of the system can be **approximated** using a small number of them. In structural analysis, shape modes associated to low frequencies are the most dangerous, and the ones from interest when studying vibration. Recall that $\mathbf{\Phi}$ and $\boldsymbol{\lambda}$ only depend on the matrices **K** and **M**, which can be computed using the static FEM program previously presented. For instance, mode shapes associated to the four lowest natural frequencies of the cantilever beam defined in 4.6 are displayed in the following figure:

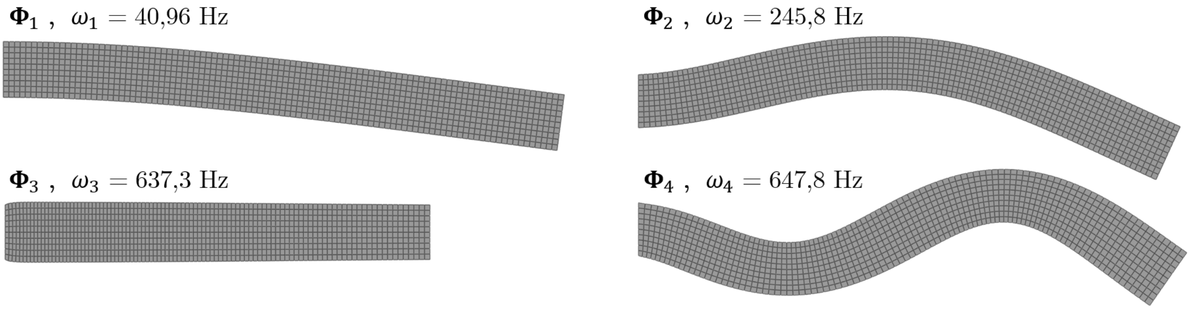

$\mathbf{\Phi}_1$, $\omega_1 = 40{,}96$ Hz
$\mathbf{\Phi}_2$, $\omega_2 = 245{,}8$ Hz
$\mathbf{\Phi}_3$, $\omega_3 = 637{,}3$ Hz
$\mathbf{\Phi}_4$, $\omega_4 = 647{,}8$ Hz

Figure 5.1: Shape modes associated to the first four lower natural frequencies of a 2D fixed-free beam

An important property of eigenvectors is that they are orthogonal with respect to **K** and **M**:

$$\mathbf{\Phi}_i^T \mathbf{M}\, \mathbf{\Phi}_j = 0\,, \quad \mathbf{\Phi}_i^T \mathbf{K}\, \mathbf{\Phi}_j = 0 \quad i \neq j \tag{5.10}$$

The previous equation can only be applied if the matrices involved in the system are symmetric and positive semidefinite, and implies that the matrix of natural modes $\mathbf{\Phi}$ diagonalizes both **K** and **M**. It is a common approach to normalise $\mathbf{\Phi}$ such that $\mathbf{\Phi}_i^T \mathbf{M} \mathbf{\Phi}_i = \mathbf{I}$, which results into $\mathbf{\Phi}_i^T \mathbf{K} \mathbf{\Phi}_i = \boldsymbol{\omega}^2$. Introducing the generalised coordinates $\mathbf{q} \in \mathbb{R}^n$ such that $\mathbf{d} = \mathbf{\Phi}\, \mathbf{q}$, and pre-multiplying by $\mathbf{\Phi}_i^T$ equation 5.5:

$$\overbrace{\mathbf{\Phi}^T\, \mathbf{M}\, \mathbf{\Phi}}^{\mathbf{I}}\; \ddot{\mathbf{q}} + \overbrace{\mathbf{\Phi}^T \mathbf{K}\, \mathbf{\Phi}}^{\boldsymbol{\omega}^2}\; \mathbf{q} = 0 \quad \Rightarrow \quad \ddot{\mathbf{q}} + \boldsymbol{\omega}^2\, \mathbf{q} = 0 \tag{5.11}$$

The above expression is, in fact, a system of uncoupled differential equations subject to the following initial conditions:

$$\mathbf{q}_{(0)} = \mathbf{q}^0 = \mathbf{\Phi}^T \mathbf{M}\, \mathbf{d}^0 \;\; , \quad \dot{\mathbf{q}}_{(0)} = \dot{\mathbf{q}}^0 = \mathbf{\Phi}^T \mathbf{M}\, \dot{\mathbf{d}}^0 \tag{5.12}$$

Recalling the definition of **q** stated in equation 5.6, the exact solution of the system can be expressed in uncoupled form as:

$$\mathbf{d} = \sum_{i=1}^{n} \mathbf{\Phi}_i \left(\mathbf{q}_i^0 \cos \omega_i\, \mathrm{t} + \frac{\dot{\mathbf{q}}_i^0}{\omega_i} \sin \omega_i\, \mathrm{t} \right) \tag{5.13}$$

In order to postprocess the results, displacements of every degree of freedom are to be computed at different values of time and joined together into a matrix. The time between two consecutive samples is called **time-step**, and in order to capture the vibration phenomena it has to be less than half the period of the highest frequency of interest. Once displacements are computed, they can be animated through **GID**.

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_{(t_1)} & \mathbf{d}_{(t_2)} & \cdots & \mathbf{d}_{(T)} \end{bmatrix} = \begin{pmatrix} d_{1\,(t_1)} & d_{1\,(t_2)} & \cdots & d_{1\,(T)} \\ d_{2\,(t_1)} & d_{2\,(t_2)} & \cdots & d_{2\,(T)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n\,(t_1)} & d_{n\,(t_2)} & \cdots & d_{n\,(T)} \end{pmatrix} \tag{5.14}$$

As it has been pointed before, sometimes it is enough with an approximation of the solution, obtained using only **m** < **n** terms. A good strategy to asses which shape modes are the most relevant for a given structure is to use the **normalised initial amplitude** associated to each mode as guide:

$$\bar{\mathbf{q}}_i^0 = \left| \frac{\mathbf{q}_i^0}{\|\mathbf{q}_i^0\|} \right| \tag{5.15}$$

For the presented 2D beam, normalised initial amplitudes are plotted in a bar graph in figure 5.2 for the six shape modes with lower frequency. It can be seen that the most excited mode shape is the one corresponding to pure bending ($\boldsymbol{\Phi}_1$), which makes sense since the beam is under the effect of a distributed load on its upper side. Notice that in order to find this out, **initial conditions** need to be defined. Different modes can dominate the same structure under distinct initial conditions.

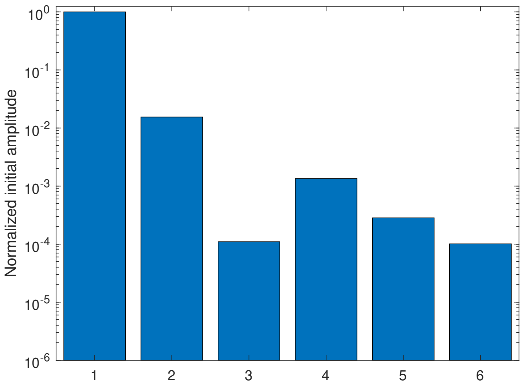


Figure 5.2: Normalised initial amplitude versus mode shape number for a 2D fixed-free beam under the effect of a distributed load

5.1.2. Damped free vibrations

Modal analysis can handle damped systems only if some requirements regarding the damping matrix are met: **D** needs to be either diagonal or a linear combination of **M** and **K**. The last assumption is known as Rayleigh damping, and has already been presented in equation 3.15. The equation describing a system under Rayleigh damping is written as:

$$\mathbf{M}\,\ddot{\mathbf{d}} + \mathbf{D}\,\dot{\mathbf{d}} + \mathbf{K}\,\mathbf{d} = 0 \;\Rightarrow\; \mathbf{M}\,\ddot{\mathbf{d}} + \left(\bar{\alpha}\,\mathbf{M} + \bar{\beta}\,\mathbf{K}\right)\dot{\mathbf{d}} + \mathbf{K}\,\mathbf{d} = 0 \tag{5.16}$$

Introducing the change of variable $\mathbf{d} = \boldsymbol{\Phi}\,\mathbf{q}$, and pre-multiplying by $\boldsymbol{\Phi}_i^T$:

$$\overbrace{\boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi}}^{\mathbf{I}}\ddot{\mathbf{q}} + (\bar{\alpha}\,\overbrace{\boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi}}^{\mathbf{I}} + \bar{\beta}\,\overbrace{\boldsymbol{\Phi}^T\mathbf{K}\,\boldsymbol{\Phi}}^{\boldsymbol{\omega}^2})\,\dot{\mathbf{q}} + \overbrace{\boldsymbol{\Phi}^T\mathbf{K}\,\boldsymbol{\Phi}}^{\boldsymbol{\omega}^2}\mathbf{q} \;\Rightarrow\; \ddot{\mathbf{q}} + \left(\bar{\alpha} + \bar{\beta}\boldsymbol{\omega}^2\right)\dot{\mathbf{q}} + \boldsymbol{\omega}^2\mathbf{q} = \mathbf{0} \tag{5.17}$$

Which is a system of uncoupled differential equations. Defining the damping ratio associated to the i—th eigenvalue $\bar{\xi}_i$ and the damped oscillation frequency $\overline{\omega_i}$:

$$\bar{\xi}_i = \frac{1}{2\omega_i}\left(\bar{\alpha} + \bar{\beta}\omega_i^2\right) \quad , \quad \overline{\omega}_i = \omega_i\sqrt{1 - \bar{\xi}_i^2} \tag{5.18}$$

The system of equations can be decoupled into:

$$\ddot{\mathbf{q}}_i + 2\overline{\xi}_i\,\omega_i\,\dot{\mathbf{q}}_i + \omega_i^2\,\mathbf{q}_i = 0 \;\;,\quad i = 1,2\ldots n \tag{5.19}$$

Up to this point, two different situations may occur depending on the value of $\overline{\xi}_i$. If $0 \le \overline{\xi}_i < 1$, then $\overline{\omega}_i$ is real valued and the exact expression for the generalised coordinates is:

$$\mathbf{q}_i = e^{-\overline{\xi}_i\omega_i\mathrm{t}}\left(\mathbf{q}_i^0\cos\overline{\omega}_i\mathrm{t} + \frac{\dot{\mathbf{q}}_i^0 + \overline{\xi}_i\omega_i\,\mathbf{q}_i^0}{\overline{\omega}_i}\sin\overline{\omega}_i\mathrm{t}\right) \tag{5.20}$$

This response is known as **subcritical oscillation**, and can be interpreted as an undamped free vibration whose amplitude decreases exponentially in time. On the other hand, if $\overline{\xi}_i > 1$, damped oscillation frequencies are no longer real valued and generalised coordinates are expressed as:

$$\mathbf{q}_i = \mathbf{q}_i^0\; e^{-\omega_i\left(\overline{\xi}_i+\sqrt{\overline{\xi}_i^2-1}\right)\mathrm{t}} \tag{5.21}$$

This is a non-oscillating response known as **overcritical motion**. In both cases, physical displacements can be computed in the same way as in the undamped case:

$$\mathbf{d} = \sum_{i=1}^{n}\mathbf{\Phi}_i\,\mathbf{q}_i \tag{5.22}$$

5.1.3. Modal analysis of rotating structures

The presented method of modal decomposition can be applied to the case of study of rotating structures, but subject to some limitations. The approach that has been developed previously is only valid if the matrices involved in the system are **constant**. Recall from equation 3.13 that for rotation problems, the stiffness and damping matrices are the addition of a static component, which is constant, and a rotating one that depends on angular velocity and acceleration. The latter varies with time if the angular acceleration is different from zero. Thus, the only possibility for the system to be tackled through modal analysis is to rotate under **constant angular velocity** Ω. In that case, the eigenvalue problem is written as:

$$\left(\mathbf{K}_{\text{static}} + \mathbf{K}_{\text{rot}} - \boldsymbol{\omega}^2\,\mathbf{M}\right)\mathbf{\Phi} = 0 \tag{5.23}$$

And if the problem is either 1D or 2D:

$$\left(\mathbf{K}_{\text{static}} - \Omega^2\mathbf{M}\; - \boldsymbol{\omega}^2\,\mathbf{M}\right)\mathbf{\Phi} = 0 \;\;\Rightarrow\;\; \mathbf{K}_{\text{static}} - (\Omega^2\; + \boldsymbol{\omega}^2)\,\mathbf{M} = 0 \tag{5.24}$$

The above expression unfolds the **softening effect** predicted in 3.17, which produces a decrease in the natural frequencies as angular velocity increases. The relationship between natural frequency of the static and rotating case is:

$$\boldsymbol{\omega}_{\text{rot}} = \sqrt{\boldsymbol{\omega}_{\text{static}}^2 - \Omega^2} \tag{5.25}$$

According to the above equation, lower frequencies are the most affected by the softening effect. This behaviour can be represented through a Campbell diagram (see figure 5.3), where natural frequencies of a rotating structure are plotted as a function of the angular speed. This softening effect, however, is rarely seen in real scenarios: it is a consequence of the **limitations** of the model, which is not able to account for a counteracting stiffening effect. The decrease of the natural frequencies emerges in fact due to the non-inertial character of the frame of study.

Supposing that the 2D beam studied in figure 5.1 undergoes a rotation at constant angular velocity, then the natural frequencies linked with the presented mode shapes would decrease as the spinning speed grows in the way it is displayed in the following graph:

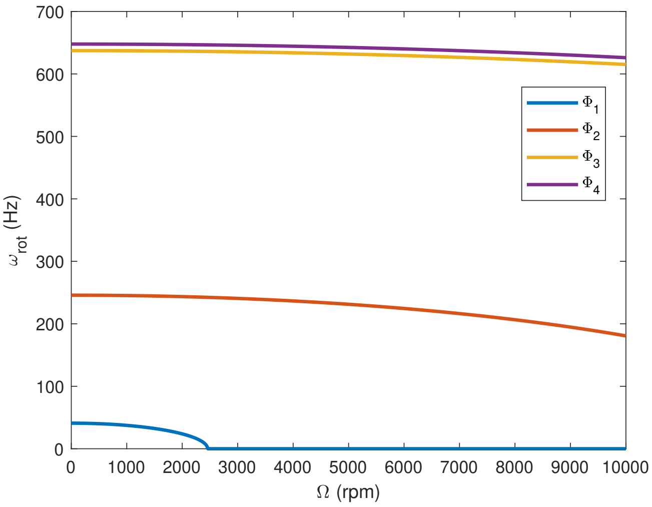


Figure 5.3: Campbell diagram representing the natural frequencies of a 2D beam rotating at constant speed

5.1.4. Limitations

Until now, only stationary scenarios have been considered as modal analysis is not capable of accounting the effect of angular acceleration. Another drawback of this method is related with the requirement that **D** needs to be a linear combination of **K** and **M**. This is not the case of rotating structures, as under the effect of rotation, the damping matrix adopts the form:

$$\mathbf{D} = \mathbf{D}_{\text{static}} + \mathbf{D}_{\text{rot}} = \overline{\alpha}\,\mathbf{M} + \overline{\beta}\,\mathbf{K}_{\text{static}} + \mathbf{D}_{\text{rot}} \tag{5.26}$$

Where $\mathbf{D}_{\text{rot}}$, that accounts for the Coriolis effect, cannot be expressed in terms of **K** nor **M** and neither is a symmetric matrix. Therefore, modal analysis in rotating structures becomes impracticable unless **Coriolis effect** is neglected. Nonetheless, it is a powerful tool that allows extraction of important information about the system, such as natural frequencies and mode shapes. Moreover, modal decomposition leads to exact results if all the **n** terms of the summation are taken into consideration. This exact solution will help assessing the validity of the approximated results obtained through the numerical methods presented in the next section.

5.2. Numerical integration

In order to overcome the limitations modal analysis is subject to, numerical integration methods arose. The key is to solve the elastodynamic differential equation by integrating over the time domain. As these integrals do not have exact close solution, they are approximated trough a set of numerical methods and algorithms. In particular, the implicit **Newmark *β*-method** [12] is the one covered in this report. To start with, the time interval is divided into N time steps:

$$[0,\mathrm{T}] = \left[\mathrm{t}^0,\mathrm{t}^1\right] \cup \left[\mathrm{t}^1,\mathrm{t}^2\right] \ldots \cup \left[\mathrm{t}^n,\mathrm{t}^{n+1}\right] \ldots \left[\mathrm{t}^{N-1},\mathrm{t}^N\right] \tag{5.27}$$

Where t^n is a generic time value. Following this description, equation 5.3 is discretised:

$$\mathbf{M}\,\mathbf{a}^{n+1} + \mathbf{D}^{n+1}\,\mathbf{v}^{n+1} + \mathbf{K}^{n+1}\,\mathbf{d}^{n+1} = \mathbf{F}^{n+1} \quad \text{where} \quad \mathbf{a} = \ddot{\mathbf{d}}\ ,\ \ \mathbf{v} = \dot{\mathbf{d}} \tag{5.28}$$

Recall that **D** and **K** are no longer constant, as angular acceleration is now taken into consideration. Newmark β-methods are a family of integration schemes in which displacements and velocities at time t^{n+1} are uploaded through the following formulae:

$$\textbf{Initial conditions} \;\Rightarrow\; \mathbf{a}^0 = \mathbf{M}^{-1}\left(\mathbf{F}^0 - \mathbf{D}^0\,\mathbf{v}^0 - \mathbf{K}^0\,\mathbf{d}^0\right) \tag{5.29}$$

$$\mathbf{d}^{n+1} = \tilde{\mathbf{d}}^n + \beta\,\Delta\mathrm{t}^2\,\mathbf{a}^{n+1}\ ,\ \ \text{with}\ \ \tilde{\mathbf{d}}^n = \mathbf{d}^n + \Delta\mathrm{t}\,\mathbf{v}^n + \frac{\Delta\mathrm{t}^2}{2}(1-2\beta)\,\mathbf{a}^n \tag{5.30}$$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^n + \gamma\,\Delta\mathrm{t}\,\mathbf{a}^{n+1}\ ,\ \ \text{with}\ \ \tilde{\mathbf{v}}^n = \mathbf{v}^n + (1-\gamma)\Delta\mathrm{t}\,\mathbf{a}^n \tag{5.31}$$

Where $\Delta\mathrm{t} = \mathrm{t}^{n+1} - \mathrm{t}^n$ is the time step and β and γ are numerical parameters. The parameter γ introduces damping to the numerical method, controlling what is called **artificial viscosity** [12]. It is used to reduce noise in the solution. If $\gamma = \frac{1}{2}$, no damping is added, whereas if $\gamma > \frac{1}{2}$ the artificial viscosity added to the system is proportional to $\gamma - \frac{1}{2}$. On the other hand, the β parameter defines how much implicit or explicit the method is. Continuing with the formulation of the numerical scheme, if equation 5.31 is introduced into 5.28:

$$(\mathbf{M} + \gamma\,\Delta\mathrm{t}\,\mathbf{D}^{n+1})\,\mathbf{a}^{n+1} + \mathbf{K}^{n+1}\,\mathbf{d}^{n+1} = \mathbf{F}^{n+1} - \mathbf{D}^{n+1}\,\tilde{\mathbf{v}}^n \tag{5.32}$$

Acceleration at t^{n+1} is obtained from equation 5.30:

$$\mathbf{a}^{n+1} = \frac{1}{\beta\,\Delta\mathrm{t}^2}\left(\mathbf{d}^{n+1} - \tilde{\mathbf{d}}^n\right) \tag{5.33}$$

Then, expression 5.32 transforms into:

$$(\tilde{\mathbf{M}}^{n+1} + \mathbf{K}^{n+1})\,\mathbf{d}^{n+1} = \mathbf{F}^{n+1} - \mathbf{D}^{n+1}\,\tilde{\mathbf{v}}^n + \tilde{\mathbf{M}}^{n+1}\,\tilde{\mathbf{d}}^n\ ,\ \ \text{with}\ \ \tilde{\mathbf{M}}^{n+1} = \frac{\mathbf{M} + \gamma\Delta\mathrm{t}\,\mathbf{D}^{n+1}}{\beta\,\Delta\mathrm{t}^2} \tag{5.34}$$

The algorithm for the implicit case ($\beta > 0$) can be summarised as:

1. Compute $\tilde{\mathbf{d}}^n$ and $\tilde{\mathbf{v}}^n$ through equations 5.30 and 5.31.
2. Obtain $\mathbf{d}^{n+1}$ solving the system of equations stated in 5.34.
3. Update accelerations $\mathbf{a}^{n+1}$ and velocities $\mathbf{v}^{n+1}$ using expressions 5.33 and 5.31.

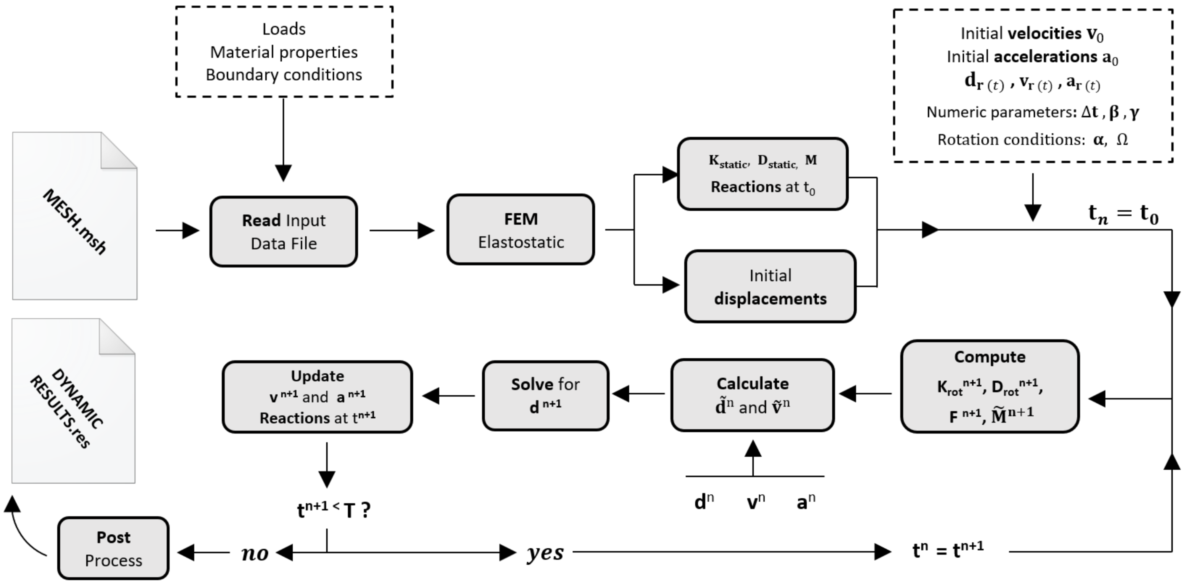

Loads
Material properties
Boundary conditions
Initial velocities $\mathbf{v}_0$
Initial accelerations $\mathbf{a}_0$
$\mathbf{d}_{\mathbf{r}\,(t)}$, $\mathbf{v}_{\mathbf{r}\,(t)}$, $\mathbf{a}_{\mathbf{r}\,(t)}$
Numeric parameters: $\Delta\mathbf{t}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$
Rotation conditions: $\boldsymbol{\alpha}$, Ω
MESH.msh
Read Input Data File
FEM Elastostatic
$\mathbf{K}_{static}$, $\mathbf{D}_{static}$, $\mathbf{M}$
Reactions at t_0
Initial displacements
$\mathbf{t}_n = \mathbf{t}_0$
DYNAMIC RESULTS.res
Update $\mathbf{v}^{n+1}$ and $\mathbf{a}^{n+1}$ Reactions at t^{n+1}
Solve for $\mathbf{d}^{n+1}$
Calculate $\tilde{\mathbf{d}}^n$ and $\tilde{\mathbf{v}}^n$
Compute K_{rot}^{n+1}, D_{rot}^{n+1}, F^{n+1}, $\tilde{\mathbf{M}}^{n+1}$
$\mathbf{d}^n$ $\mathbf{v}^n$ $\mathbf{a}^n$
$t^{n+1} < T$?
Post Process
no
yes
$t^n = t^{n+1}$

Figure 5.4: Newmark β-method algorithm schematisation for the case of rotating structures.

This scheme is found to be **unconditionally stable** for $\beta \geq \frac{\gamma}{2} \geq \frac{1}{4}$. It is found that $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ yield to reasonable convergence rates. If those particular values of numeric parameters are chosen, then the scheme is known as constant average acceleration method. Recall that if $\mathbf{F}^{n+1} = f(\mathbf{d}^{n+1}, \mathrm{t}^{n+1})$, then 5.34 would be a nonlinear system of equations, which is to be tackled through numerical schemes such as Newton-Raphson. Fortunately, this is not the case of study and displacements are simply computed as:

$$\mathbf{d}^{n+1} = \left(\tilde{\mathbf{M}}^{n+1} + \mathbf{K}^{n+1}\right)^{-1} \left(\mathbf{F}^{n+1} - \mathbf{D}^{n+1}\,\tilde{\mathbf{v}}^{n} + \tilde{\mathbf{M}}\,\tilde{\mathbf{d}}^{n}\right) \tag{5.35}$$

5.2.1. Timestep selection

The presented method is a practical tool that allows the solving of complex nonlinear systems. However, convergence of the solution is only granted for sufficient small values of Δt. Its value is directly linked with the highest frequency involved in the problem. According to the **Nyquist-Shannon** theorem, sampling frequency f_s needs to be at least twice the maximum frequency of the signal. In the case of study, two different phenomena are to be taken into account: **vibration** in the noninertial frame and **rotation** in the inertial one.

The period of the rotation motion, supposing constant angular acceleration α and initial velocity v_0, can be posed as:

$$\text{period} = \frac{1}{\alpha}\left(\sqrt{v_0 + 4\pi\,\alpha\,(k+1)} - \sqrt{v_0 + 4\pi\,\alpha\,k}\right) \;, \quad k = \text{floor}\left[\frac{1}{2\pi}(v_0 + \frac{1}{2}\alpha\,\mathrm{t}^2)\right] \tag{5.36}$$

As rotations undergoing angular acceleration for a long period of time are not usual, let's consider a rotation at very high constant spin speed, say $\Omega = 15000$ rpm. In consequence, the sampling frequency should be greater than 500 Hz, and the timestep smaller than 2 ms. To put these numbers in context, recall natural frequencies from figure 5.1: the third mode presented a frequency of 637 Hz, implying a sampling frequency greater than 1275 Hz.

In general, vibration frequencies are higher than the rotation velocity, and thus the selection of Δt will not be influenced by the velocity of the rotation motion. The criteria to follow when picking a sampling frequency is to ensure that it is at least twice the highest vibration frequency to capture.

5.2.2. Validation

In order to asses whether the numerical scheme is performing properly, numerical results will be compared with those obtained with modal analysis. As the latter is restricted to free undamped static structures, the case of study will involve the free vibration of the 2D cantilever rectangular beam presented in 4.7. Imagine the beam under the effect of a distributed load along its upper side, when all of the sudden the force **vanishes** and the structure starts to oscillate from its deformed configuration around its equilibrium position.

Although highly visual, animations are not a proper way of comparing quantitatively the two responses. Instead, the plot of dynamic displacements at different degrees of freedom will be compared for both modal and numerical cases. In fact, the nodes whose vertical displacement will be graphed are those located at the centre and tip of the **mean line** of the beam. The modal solution will be obtained taking into account the six first mode shapes with the lowest frequency, whereas the numerical scheme will be based on the constant average acceleration method ($\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$). The selected timestep depends of the highest natural frequency to capture, which in this case is linked with the pure bending mode. Hence, $\Delta\mathrm{t} = 1\,ms$. The results are compared in the following figure, where dots represent the approximated numerical solution:

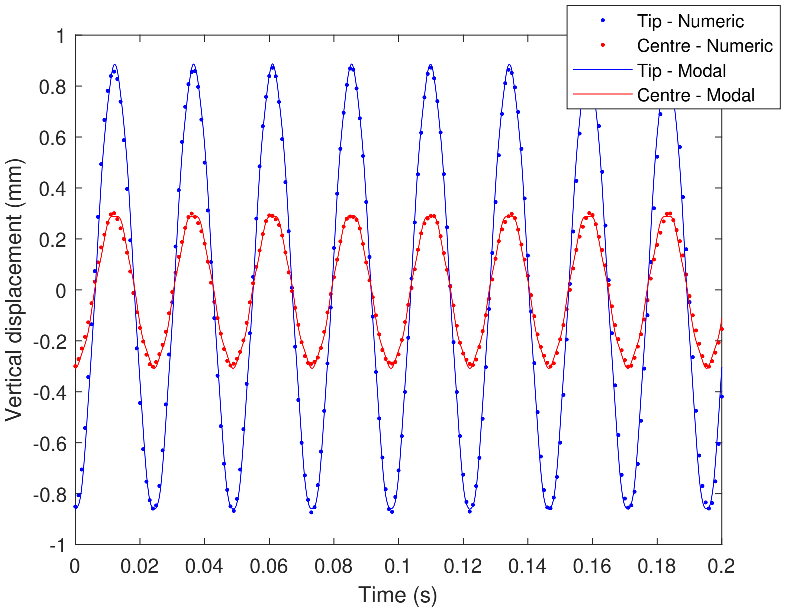


Figure 5.5: Comparison of modal and numerical methods to solve the free undamped vibration of a 2D fixed-free beam.

As can be spotted, amplitude in the tip is higher than in the centre. Furthermore, the frequency of vibration is that of the bending motion ($\sim 41\,Hz$), as was expected from figure 5.2. The main problem of comparing both solutions is that numerical methods, unlike modal analysis, cannot **filter** high frequency modes. Thus, the numeric solution is always affected by high frequency components and is not capable of representing the bending motion in isolation. In consequence, modal and numeric results will only be coincident if all the mode shapes are taken into consideration and the time step is sufficiently small. The following figure compares horizontal displacements at the tip using 600 different eigenvalues, and a numeric timestep of $0.1\mu s$:

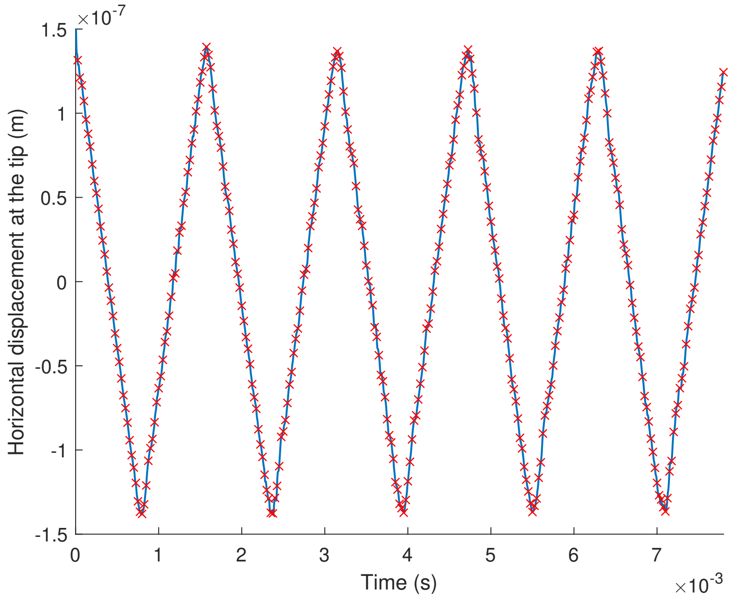


Figure 5.6: Modal and numeric (red crosses) methods comparison, this time with higher accuracy.

5.2.3. Limitations

Numerical methods allow complex and even nonlinear systems to be tackled, obtaining results with a proper degree of accuracy. Even so, one of their main drawbacks is linked with the requirement of really small time steps in order to capture structural vibrations. This implies high computation time and memory usage. Recall that in order to animate the results, displacements need to be computed for each timestep. For instance, the displacements of a 3D system with 30000 nodes studied during a timespan of $10s$ with $\Delta\text{t} = 0.1\ ms$ would take up 67 gigabytes of memory. In reality, postprocessing tools also require from other inputs, increasing even more the memory consumption. This makes numeric simulations of complex structures only feasible for institutions with high computational resources.

Another disadvantage of numerical schemes with respect to modal analysis is that no mode shapes nor natural frequencies are obtained. In fact, what is computed through numerical methods is the temporal evolution of the individual displacements at each degree of freedom. The whole phenomena of vibration is difficult to capture by only looking at individual graphs, so postprocessing tools use this information to create an animation of the dynamic response. But animations are not an outright solution, as no information regarding the predominating modes of the structure and its frequency are obtained.

When very different frequencies play an important role in the system, the latter has to be studied at different **time scales** as there will be motions with very long period coexisting with other ones which elapse very fast. The study of both motions at the same time based on animations is very tricky, and requires from patience and experience. In order to overcome this limitation, the next section reviews a promising method that captures the predominating modes and frequencies of a structure using only the raw data obtained from the numerical approach.

5.3. Singular value decomposition

The singular value decomposition, commonly referred as SVD, is one of the most versatile tools of linear algebra. It allows the factorisation of a rectangular matrix **A** of the form:

$$\mathbf{A} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^T\ , \quad \text{or} \quad \mathrm{a_{ij}} = \sum_{k=1}^{n} \mathrm{u_{ik}}\,\sigma_k\,\mathrm{v_{jk}} \tag{5.37}$$

Where:

- **A** is a m × n matrix.
- **U** is a m × m orthonormal matrix known as the matrix of left side vectors (LSV).
- **V** is a n × n orthonormal matrix known as the matrix of right side vectors (RSV).
- $\boldsymbol{\Sigma}$ is a m × n diagonal matrix known as the matrix of singular values.

The diagonal entries of $\boldsymbol{\Sigma}$ are referred as singular values σ_i, fulfilling that $\sigma_i > 0$. Quite often, the matrix of singular values is sorted in ascending order, so that $\sigma_{i+1} \leq \sigma_i$. If **A** is a real matrix, then both **U** and **S** are orthonormal: $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$. An intuitive approach to SVD is to think of **A** as a linear transformation. Then, the singular value decomposition can be though as a composition of three geometrical transformations: a rotation **U**, a scaling or stretching $\boldsymbol{\Sigma}$ and another rotation $\mathbf{V}^T$.

The SDV can be interpreted as a generalisation of the eigendecomposition, which is restricted to positive semidefinite matrices. In fact, the SVD can be split into a pair of eigenproblems:

$$\begin{aligned} \mathbf{A}\,\mathbf{A}^T = \mathbf{U}\boldsymbol{\Sigma}\overbrace{\mathbf{V}^T\mathbf{V}}^{\mathbf{I}}\boldsymbol{\Sigma}^T\mathbf{U}^T \quad &\Rightarrow \quad \mathbf{A}\,\mathbf{A}^T\mathbf{U} = \mathbf{U}\boldsymbol{\Sigma}^2 \\ \mathbf{A}^T\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}^T\overbrace{\mathbf{U}^T\mathbf{U}}^{\mathbf{I}}\boldsymbol{\Sigma}\mathbf{V}^T \quad &\Rightarrow \quad \mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{V}\boldsymbol{\Sigma}^2 \end{aligned} \tag{5.38}$$

Hence, **U** can be thought as the set of orthonormal eigenvectors of $\mathbf{A}\,\mathbf{A}^T$ and **U** as the set of orthonormal eigenvectors of $\mathbf{A}^T\,\mathbf{A}$. The singular values of **A** are in fact the square root of the eigenvalues of both $\mathbf{A}\,\mathbf{A}^T$ and $\mathbf{A}^T\,\mathbf{A}$. This property brings to light the capacity of the SVD to discover some of the same kind of information as with the **modal analysis**. In fact, the simplest numerical scheme to tackle the SVD problem consists in solving one of the above eigenproblems, obtaining either **U** or **V** as eigenvectors and $\boldsymbol{\Sigma}^2$ as eigenvalues. Then, the remaining matrix of vectors is obtained using equation 5.37.

5.3.1. Truncated SVD

The SVD is an interdisciplinary technique used in diverse fields such as signal processing, pattern recognition, data compression, weather prediction and social-media algorithms among many others. A popular application is related with **dimensionality reduction**: a process through which a given matrix **A** is approximated by a lower rank matrix $\mathbf{A}^{(r)}$. The SVD provides the best k-rank approximation to **A** of the form:

$$\mathbf{A} \approx \mathbf{A}^{(r)} = \sum_{i=1}^{r} \mathbf{U}_i\,\sigma_i\,\mathbf{V}_i \tag{5.39}$$

The key is to perform a SVD picking only the r-largest values of σ_i. If $r < n$, then the dimension of the matrices involved in the SVD problem are reduced: **U** becomes m × r, $\mathbf{V}^T$ is now r × m and $\boldsymbol{\Sigma}$ is reduced into a r × r matrix. The truncated version of the SVD is tackled as an optimization of the Frobenius norm of the difference between the original matrix and its approximation. Numerical schemes use a given value of tolerance to limit the truncation error.

Even considering the truncated model, computing the SVD can be extremely time-consuming for large-scale problems. One of the most used techniques to speed up the algorithm relies on block matrix decomposition. The factorisation no longer processes the whole matrix, but performs successive SVDs over smaller sub-matrices. Moreover, in this way the quality of the approximation can be controlled blockwise. These kind of method is known as **partitioned** truncated SVD.

To improve even more the performance of the algorithm, **randomised** techniques [10] are used to compute an approximation to the range of **A**. These techniques are of special interest when the data comes from discretization of continuum systems. An intuitive approach is to estimate the range of **A** by picking a random collection of vectors of the matrix and examining the subspace formed by the action of **A** on each of them. For sure, randomised SVD algorithms are not that simple and require a deeper knowledge on linear algebra.

5.3.2. SVD in structural analysis

In the case of study, which is structural analysis, it is from interest to work in the basis defined by the mass matrix **M** instead of the euclidean space. The goal is to re-express **A** in such a way that redundancies are filtered out and actual dominant displacement fluctuation patterns are revealed [13]. In order to develop this normalisation, **M** needs to be a symmetric positive definite matrix such that admits the Cholesky factorisation:

$$\mathbf{M} = \overline{\mathbf{M}}^T \, \overline{\mathbf{M}} \tag{5.40}$$

Where $\overline{\mathbf{M}}$ is an upper triangular matrix. Then, the truncated SVD is not performed to the matrix $\mathbf{A}$ but to the normalised matrix $\overline{\mathbf{A}} = \overline{\mathbf{M}}\,\mathbf{A}$:

$$\overline{\mathbf{A}} = \overline{\mathbf{U}}\boldsymbol{\Sigma}\mathbf{V}^T \quad \Rightarrow \quad \overline{\mathbf{A}}^{(r)} \approx \sum_{i=1}^{r} \overline{\mathbf{U}}_i \, \sigma_i \, \mathbf{V}_i \tag{5.41}$$

Finally, the right side vectors are obtained from $\overline{\mathbf{U}}$:

$$\mathbf{U} = \overline{\mathbf{M}}^{-1} \, \overline{\mathbf{U}} \tag{5.42}$$

Following this approach, the SVD can be posed in terms of structural analysis. The matrix $\mathbf{A}$ is nothing else but the matrix of displacements $\mathbf{d}$ defined in equation 5.14. Each column contains the displacements of the unconstrained degrees of freedom at a given value of time, whereas each row accounts for the temporal evolution of the displacement at a single degree of freedom. Hence, defining $\mathrm{n}_{\mathbf{l}}$ as the number of unconstrained degrees of freedom and $\mathrm{n}_{\Delta\mathrm{t}}$ as the number of time-steps into which the time domain is split, $\mathbf{A}$ is a $\mathrm{n}_{\mathbf{l}} \times \mathrm{n}_{\Delta\mathrm{t}}$ matrix.

From the structural analysis perspective, $\mathbf{U}$ is a $\mathrm{n}_{\mathbf{l}} \times \mathrm{r}$ matrix containing in each column a certain combination of nodal displacement patterns which are predominating in the system as time passes by. These can be thought, in analogy with modal decomposition, as the principal modes of the structure. In reality, left side vectors are not modal mode shapes but a certain combination of the sadistically reigning ones. Hence, on now on the vectors $\mathbf{U}_i$ will be referred as **singular modes**.

On the other hand, $\mathbf{V}^T$ is a $\mathrm{r} \times \mathrm{n}_{\Delta\mathrm{t}}$ matrix each row of which contains a statistically predominant temporal evolution of the whole structure. In contrast with modal analysis, each $\mathbf{V}_i$ describes a vibration motion which is not necessarily harmonic. The vectors $\mathbf{V}_i$ will referred as **singular oscillations**.

The grace of this method is that each $\mathbf{V}_i$ is linked with $\mathbf{U}_i$, allowing a certain intuition on how each of the singular modes evolves in time. Through an adequate analysis, one can find out the predominating frequencies of each singular oscillation (RSV), and consequently the vibration frequencies $\overline{\boldsymbol{f}}_i$ associated to each singular mode (LSV). Moreover, singular values σ_i give a hint on the intensity of each singular mode, allowing the truncated study of the system accounting only for the most relevant ones. The individual motions can then be animated using **GID**, obtaining a more clear sight of the phenomena than the one obtained with numerical integration.

As it has been pointed out, singular oscillations are no longer a pure harmonic and thus may contain the contribution of more than one natural frequency. In order to extract information regarding predominant frequencies, two numerical tools can be used. The most intuitive approach is based on looking for the **local maxima** of $\mathbf{V}_i$, and approximating the natural frequency as the average of the time intervals between successive maximums. The main drawback of the latter method is that if more than two frequencies are involved in the temporal description of the motion, only the highest one can be extracted from $\mathbf{V}_i$. Furthermore, the induced error on the measure of the latter is no longer negligible.

A more feasible approach consists in applying the **discrete Fourier transform** on $\mathbf{V}_i$ using inbuilt MATLAB functions. By doing so, the singular oscillation is mapped into the frequency domain. Predominating frequencies of each singular mode can be found by looking for the peaks on the power spectral density estimation constructed from the Fourier transform. Recall that the obtained vectors $\mathbf{U}_i$ and $\mathbf{V}_i$ have been scaled — in space and temporal coordinates, respectively — during the SVD, and thus information regarding the sampling frequency used during the numeric integration is needed to recover the actual frequencies involved in the vibration motion.

For sure, the development of a numerical algorithm capable of handling the truncated SVD problem in a short amount of time is far out of the scope of this report. Instead, a MATLAB tool developed by prof. *Hernández J.* will be used to perform a partitioned randomised SVD. This software limits the number of terms of the truncated approximation $\mathbf{A}^{(r)}$ by using a value of tolerance introduced by the user.

5.3.3. Validation

In order to determine whether the singular value decomposition is a suitable technique for structural analysis, its performance will be assessed by comparing the singular modes and predominant frequencies with the results obtained through modal analysis. Again, the case of study involves a **2D cantilever beam** in a free undamped oscillation, as covered in 5.5. Let's consider the dynamic displacements obtained from modal decomposition as the **input** data matrix for the SVD. The goal is to check whether this method is able to recover the information regarding natural modes and frequencies properly.

In this particular case, the modal response has been approximated using only the six natural modes with the lowest frequency. The sampling frequency used to discretise equation 5.13 is 500000 Hz. On the other hand, the truncated SVD will only account for the six more relevant singular modes. To help comparison with modal analysis, one may define the **normalised singular values** as

$$\overline{\sigma}_i = \frac{\sigma_i}{\sum_{i=1}^{r} \sigma_i} \tag{5.43}$$

With this in mind, the SVD and modal results are compared in the following table:

Modal Analysis	Natural frequency (Hz)	40,96	245,80	637,26	647,79	1177,99	1797,92
	Normalised initial amplitude (x 10^{-2})	99,9	1,54	0,011	0,13	0,028	0,010
SVD	Predominant frequency (Hz)	40,96	245,82	635,08	647,82	1177,98	1798,18
	Normalised singular value (x 10^{-2})	98,3	1,56	0,007	0,13	0,028	0,010

Table 5.1: Frequencies and intensities comparison: SVD vs. modal analysis

The above table reveals an outstanding degree of accuracy in the frequency determination using the SVD. The singular modes associated to the four lower predominant vibration frequencies are displayed in the following figure:

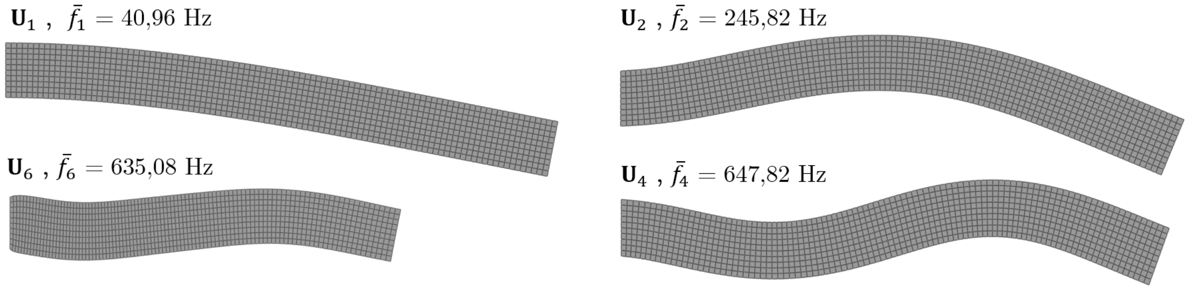

$\mathbf{U}_1$, $\bar{f}_1 = 40{,}96$ Hz
$\mathbf{U}_2$, $\bar{f}_2 = 245{,}82$ Hz
$\mathbf{U}_6$, $\bar{f}_6 = 635{,}08$ Hz
$\mathbf{U}_4$, $\bar{f}_4 = 647{,}82$ Hz

Figure 5.7: Singular modes associated to the first four lower vibration frequencies.

Comparing the above illustration with natural modes (figure 5.1), one finds out that they are recovered almost exactly. The only exception is found in the third natural mode, related with the axial deformation. In this case, neither the predominant frequency nor the singular mode match exactly with the modal results. The reason behind this discordance is that the axial mode is merely activated in a bending scenario. In consequence, being the SVD a statistical method, less reliability is to be expected when recovering the axial mode.

With all, the singular value decomposition has been proven to be a good approach in recovering relevant information from a given matrix of data. In structural analysis, it is a promising tool for obtaining the **predominant motions** when modal analysis is not practicable. The next chapter will cover the application of the SVD to a set of rotating scenarios.

6

Analysis of rotating beams using the SVD

Once the methodology for handling rotating structures in non-stationary regime is fully developed, it is time to tackle a few scenarios involving rotating machinery. The simulations will be carried taking advantage of the Newman-β algorithm presented in the previous section, and then the SVD will used to disclose the dominant modes and frequencies of the dynamic response. The results will be **compared** with those obtained through modal analysis, which is restricted to non-rotating vibrations. Hence, a hint on the effects centrifugal, tangential and Coriolis forces have on rotating structures will be brought to light in this chapter.

Postprocessing of results will be carried out by **GiD**, which uses the generic spatial basis $\{\text{xyz}\}^T$. The simulations performed in the present chapter will consider only variables in the local frame of reference $\{\varsigma\,\varrho\,\varpi\}^T$, so although some figures incorporate a set of xyz axes, keep in mind that they represent in fact the corotational frame.

6.1. Two-dimensional cantilever beam

The first case of study continues with the analysis of the rectangular 2D **cantilever beam** studied in the previous section. The beam is $0.2 \times 2\,m$ and modelled using 1000 quadrilateral elements. The main difference is that the beam now undergoes a **rotation** perpendicular to the plane, with its axis located at the centre of the fixed end of the beam. The beam is made from 6061 T6 aluminium alloy ($E = 70\,GPa$, $\nu = 0.3$), and the considered density is $\rho = 2700 kg/m^2$.

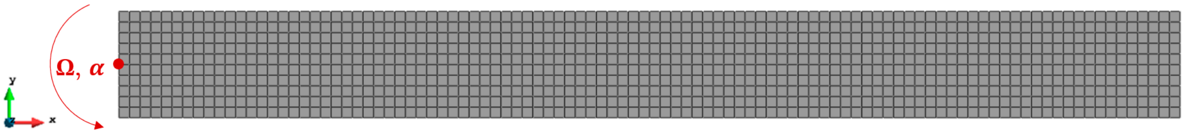


Figure 6.1: Rectangular cantilever beam subject to rotation.

The loading scenario is quite similar as the presented in 4.6: the beam is under the effect of a distributed ϱ load on its upper side (-20 kN/m), while it is **rotating** at a certain angular velocity. Initial displacements under these conditions will be computed using the static FEM software. The dynamic case will account for a **sudden** vanishing of the distributed load and the consequent vibration of the beam. This oscillating behaviour will be analysed under the effect of different angular speeds and accelerations. No internal **damping** will be considered in this section. For comparison purposes, the first eight natural frequencies and modes of the presented structure can be found in the report attachments (figure D.1.1).

6.1.1. Constant angular velocity

The first analysis will involve constant angular speed. The goal is to keep track of the vibration frequencies at different angular velocities, and compare the results with the **softening effect** predicted by modal analysis. The imposed angular speed cannot be indiscriminately high, as deformations would no longer be considered small. Using the 1D solver developed in 4.2.2, it has been found that elastic displacements reach a 1% of the length of the beam when $\Omega = 4000$ rpm. Hence, this value will not be exceeded in this analysis.

It is important to select an adequate timespan and timestep, as numerical results are sensitive on these parameters. The **timestep** is chosen depending on the maximum frequency to capture, in this case $f_{\max} \approx 2500\,Hz$, corresponding to the eighth natural frequency. Hence, $\Delta t < 0.2\,ms$. In order to gain accuracy, the used timestep will be $\Delta t = 5\,\mu s$. To keep computation time low, the **timespan** will be no longer than $0.2\,s$. With these numeric parameters, and using the constant average acceleration method, the mean time of computation is around 25 minutes. Smaller time-steps or greater time-spans would be too much time-consuming.

Campbell diagram

The dynamic analysis is to be developed for three different angular velocities: 50, 100 and 200 rad/s. The SVD is truncated to the **eight** higher singular values. The left and right side vectors obtained for each case are displayed in D.1.2. To ease visualisation of the RSVs, each one is plotted for a certain fraction of the total timespan. The dominant frequencies are found looking for the peaks of the **discrete Fourier transform** of each RSV. The obtained frequencies are presented in the following figure, and compared with the softening effect predicted by modal decomposition.

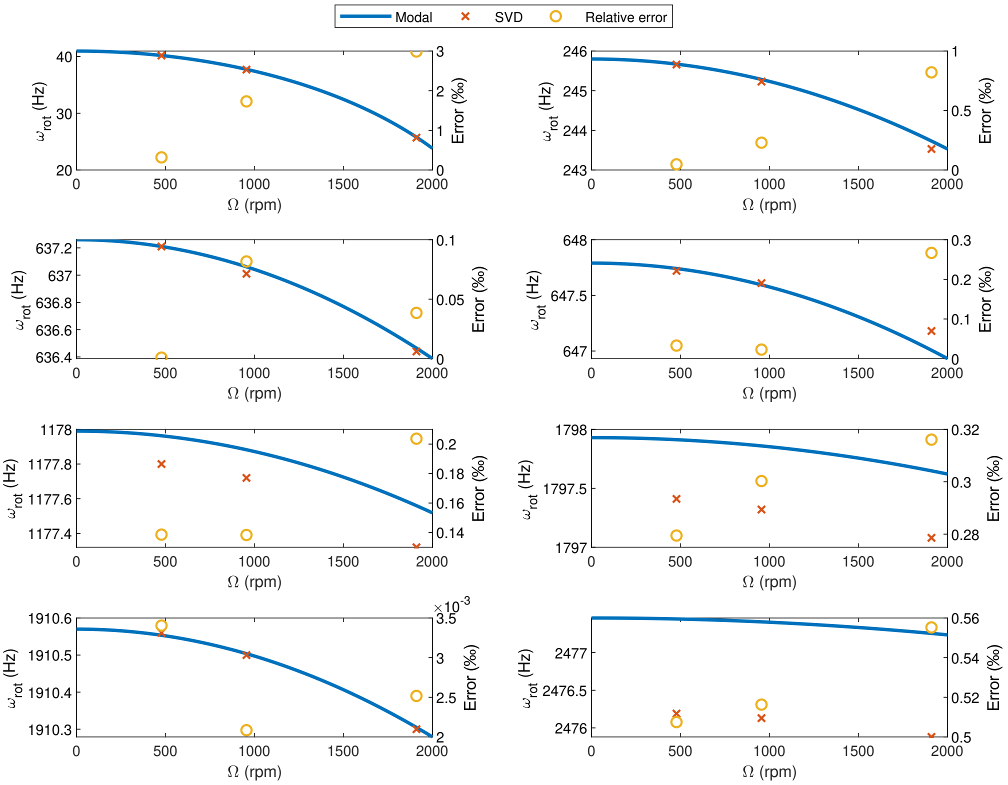


Figure 6.2: Campbell diagram: comparison between modal prediction and SVD results.

As can be spotted, in most cases the dominant frequency obtained using SVD matches almost exactly with the **predicted** natural frequency. In other cases (modes 5, 6 and 8), the values are not coincident but very similar ($\pm 1 Hz$). In all likelihood, these differences can be attributed to the **probabilistic** behaviour of the SVD, as the numeric sample has a timespan of only $0.2\,s$. Even so, in all cases the vibration frequencies predicted by the SVD decrease with the spin speed, matching with the expected softening effect. Thereby, it can be deduced that neither Coriolis nor centrifugal forces induce a **significant** change in the natural frequencies of rotating beams. In the same way, by comparing the LSVs from D.1.2 with the natural modes of the beam, it can be stated that mode shapes do not either change with the angular velocity.

Recall that the obtained results shown in D.1.2 are just **predominant** patterns of the dynamic response. Thus, the RSVs representing reigning vibrations are no longer pure harmonics, but mostly the result of a certain combination of natural frequencies. In most cases, they match almost perfectly to a sine or cosine function, and hence the predominant frequency is easy to determine. That is not the case of the RSVs associated to axial vibrations (natural modes 2 and 7 from figure D.1.1), which are the result of more than one predominant frequency. For instance, the pure axial motion has a 0 Hz frequency associated to it, caused by the offset in displacements the centrifugal force induces. To visualise the frequencies involved in the vibration phenomena, figure D.11 displays the **power spectral density** estimation of the sum of the right side vectors.

Singular values

Although Coriolis and centrifugal acceleration induce little changes in natural frequencies, they do change considerably the **intensity** in which natural modes show up. Considering the normalised singular values as the indicator of intensity, figure D.8 shows how axial modes (2 and 7) gain intensity as rotation speed increases, whereas flexural modes weaken. This effect is caused by the centrifugal force, the responsible of axial excitation, which increases proportionally to Ω^2. The numeric values of predominant frequencies and normalised singular values are presented in table D.2.

Reactions

Another interesting variable to keep track of is the **reaction torque**, which at last instance is the one determining the required torque to spin at the desired speed. For the presented scenario and studied angular speeds, the reaction torque is found to oscillate along with the beam, as displayed in figure D.9.

As the motion is not under the effect of angular acceleration, no rigid body reactions are present. Thus, figure D.9 only accounts for the elastic reactions described in 3.21. The torque is mainly induced by the effect of transverse reactions, as local horizontal reactions due to centrifugal forces produce no net torque around the specified rotation axis. Hence, the required torque is closely related with the bending modes, and thus, with its natural frequency, which decreases with Ω. In this case, reactions are not generated by a certain applied force but as a consequence of the **inertia forces** induced by the high accelerations present in the vibration phenomena.

At some intervals, the required torque becomes negative: half of the time, the beam is bending in the direction of rotation, tending to accelerate the structure. Hence, the overall work is zero. This is an **artificial scenario**: the angular speed of actual rotors is never constant, whereas the torque injected by the driving device is in fact uniform. In real cases, the vibration of the beam accelerates and decelerates the rotation motion in a small amount (as stated in 1.2). Moreover, real rotors work within viscous fluids, which act as dampers of the vibration motion.

Displacements

Regarding dynamic results, displacements at the tip of the beam will be reviewed for different rotation velocities. Displacements in the local **perpendicular** direction are induced mostly by the pure flexural mode, and hence its vibration motion is easy to capture as it is associated to a low frequency. The amplitude of these displacements does not increase significantly with Ω, but it is the decrease in frequency the most remarkable effect rotation velocity introduces. Figure D.10 presents the numeric results and their lower rank approximation computed using the truncated SVD with eight singular values.

On the other hand, **axial** displacements are highly dependant on Ω, as the centrifugal force that excites axial modes increases with the square of the angular velocity. In fact, the amplitude increases by a factor of 16 when comparing figure D.12 (a) with figure D.12 (c). Moreover, the local horizontal displacements are linked to a relatively high frequency, and thus the truncated SVD is not capable to recreate the numerical result exactly. This issue is easily overcome by decreasing the value of tolerance of the SVD.

Stresses

In structural analysis, one of the most usual parameters used to determine whether the structure is going to fail or not is the **security factor**. In this example, computation of stresses is quite straightforward, as no internal damping is considered. Stresses and strains are obtained directly from the numerical scheme, but unlike displacements, they are not passed into the SVD for dominant patterns recognition. When dealing with stresses, their distribution is not so important as it is the peak value, from which depends the security factor. Figure 6.3 displays the peak values of $\sigma_{\varsigma\varsigma}$, $\sigma_{\varrho\varrho}$ and $\tau_{\varsigma\varrho}$ as time passes by for the most favourable case ($\Omega = 50$ rad/s).

From figure 6.3, it is found that maximum stresses oscillate at twice the natural frequency of the flexural mode. That is caused by the fact that maximum stresses take place at maximum deformed configurations — highest strains — which occur two times per oscillation. For the used aluminium, the yield strength is around 276 MPa [14], and so the security factors are 50, 13, and 4 for 0, 50 and 100 rad/s, respectively.

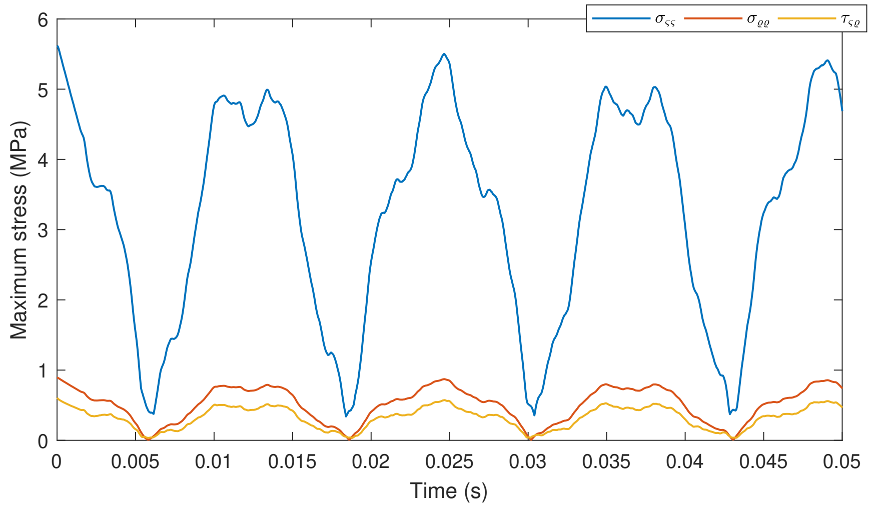


Figure 6.3: Temporal evolution of the maximum stresses (absolute value) for $\Omega = 50$ rad/s.

Results animation

To close down this section, the outputs from the dynamic analysis will be animated through **GiD**. However, postprocessing is a heavy task for computers with humble computation capabilities, as results are encoded in ASCII. For that reason, only the case corresponding to $\Omega = 50$ rad/s will be dynamically animated. To simplify the animation, the deformed configuration will be displayed in the rotating frame, and only stresses in the local horizontal direction will be represented through a colour map.

During the bending motion, one expects to found a symmetric distribution of stresses with respect to the beam's axis: one side under traction and the other under compression. Figure D.13 shows how this symmetry is **broken** in rotating scenarios, as the centrifugal force introduces an axial displacement proportional to the distance from the rotation axis that counteracts the compressing stresses.

6.1.2. Constant angular acceleration

The second and last analysis regarding the rectangular cantilever beam accounts for the effect of a constant acceleration. Recall that angular acceleration was neglected in modal analysis, as variable rotation speed is not compatible with its formulation. Hence, this analysis aims to unfold the effect angular acceleration has on rotating structures. For the sake of simplicity, only two simulations will be carried out, corresponding to $\alpha = 5$ and $50\,\mathrm{rad/s^2}$. In both cases, the initial angular velocity will be $\Omega_0 = 50$ rad/s.

The numerical parameters will be the exact same as in the case of constant spin speed. Thus, the timespan will not be large enough for the angular acceleration to introduce significant changes in angular velocity — in the case of higher acceleration, $\Delta\Omega = 10$ rad/s. For this reason, no Campbell diagram will be represented, as the softening effect induced by α will be relatively small and difficult to measure. Remember that the rotation stiffness matrix is proportional to:

$$\begin{pmatrix} -\Omega^2 & -\alpha & 0 \\ \alpha & -\Omega^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{6.1}$$

Thus, as $\Omega^2 \gg \alpha$, no significant effect is expected. The first difference with the previous case is the **tangential** force introduced by α, acting perpendicular to the beam. In consequence, the flexural mode is highly activated when angular acceleration is present. In fact, the flexural motion is so predominant that the SVD is no longer capable of extracting neither the natural frequency nor the mode of the axial vibration. Figure D.16 reveals how the second predominant mode from SVD now corresponds to a **combination** of both axial and perpendicular displacements. Intensity of axial modes (three and seven from figure D.1.1) decreases with angular acceleration, as shown in figure D.18, that compares normalised singular values of different natural modes for different values of α.

Notice that the truncated SVD is not capable of recovering the seventh natural mode when $\alpha = 50\,\mathrm{rad/s^2}$. In fact, in this case the seventh and eight LSVs are the same and correspond to the eight natural mode Φ_8. Predominant frequencies and singular values obtained are summarised in table D.3.

Displacements

Again, displacements at the tip will be considered representative of the vibration phenomena. Displacements in the perpendicular direction oscillate at the first natural frequency, but this time the vibration is not around zero but to a new equilibrium position which depends on the magnitude of the applied tangential force. The higher the angular acceleration, the more deformed the configuration at equilibrium will be. This behaviour can be spotted in figure D.19.

On the other hand, axial displacements are relatively smaller (one order of magnitude below perpendicular ones). They oscillate at two different frequencies, corresponding to the flexural and axial natural modes. It is worth pointing that, as the angular velocity increases with time, so does the centrifugal force. For that reason, the mean value of local horizontal displacements **increases** with time, as captured by the second RSV in D.17. The rate of this rise is proportional to the angular acceleration. This effect can be identified in figure 6.4, where horizontal displacements are represented. The increase in length of the beam $\Delta\varsigma$ during the analysed timespan are 0.11 and 0.01 mm for $\alpha = 50$ and 5 rad/s^2, respectively. If the angular acceleration (for instance $\alpha = 50$ rad/s^2) persisted in time, that would result in an increase of half millimetre in the beam's length every second.

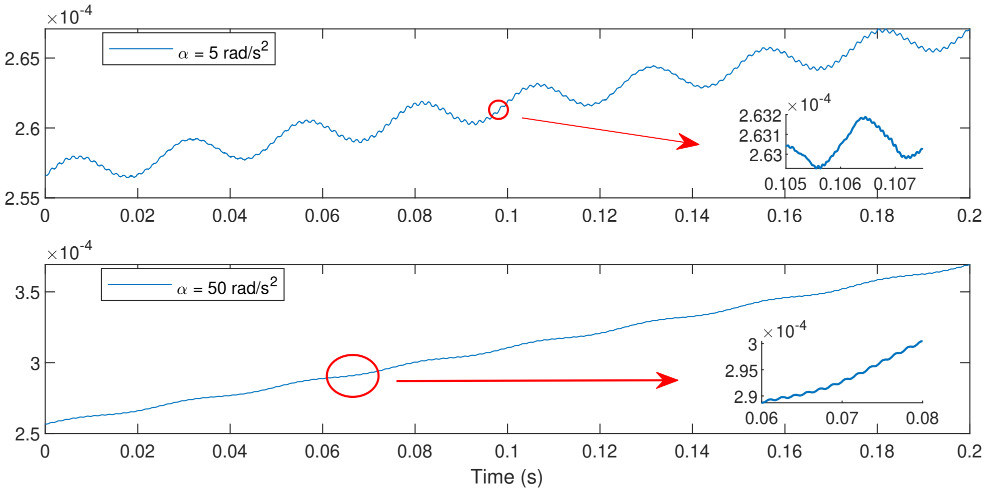


Figure 6.4: Local horizontal displacements (m) at the tip of the beam for different values of angular acceleration.

Reactions

As in the previous case, the obtained reactions are those the driving device must inject into the structure to ensure the desired acceleration. In this case of constant α, the elastic reaction torque oscillates at the bending frequency, but with a mean value different from zero as the equilibrium position of the beam is no longer the horizontal configuration. Figure D.20 displays how the mean value of elastic reaction torque increases with α.

It is worth pointing that, in the case of variable angular speed, a rigid body motion must be taken into account and added to the previous one, as stated in equation 3.2.3. In this case, the rigid body torque is constant:

$$\mathbf{W}_{rb} = \mathbf{J_z}\,\alpha = \overbrace{14436}^{kg\,m^2}\,\alpha \xrightarrow{\alpha = 50\,\mathrm{rad/s^2}} \mathbf{W}_{rb} = 721{,}8\;kN \cdot m \tag{6.2}$$

This constant value is to be added to the elastic torque from D.20, which is significantly smaller. Recall that this is not a real scenario, as the kinematics of the motion are often the unknown rather than the input parameter. The required power the driving engine must provide is given by the product of the total reaction torque (rigid body and elastic) by the instantaneous angular velocity Ω.

Stresses

In order to validate structural reliability, one needs to focus on the study of stresses, which determine the security factor. Again, as no internal damping is considered, computation of stresses is done as in the static scenario. For simplicity, only peaks values will be considered in this section. As expected, stresses increase with α, although they are most sensitive to a change in angular velocity. Figure 6.5 reveals how stresses in the local horizontal direction are the higher ones, and oscillate in the flexural natural frequency. The peaks in stresses correspond to the maximum deformed configurations.

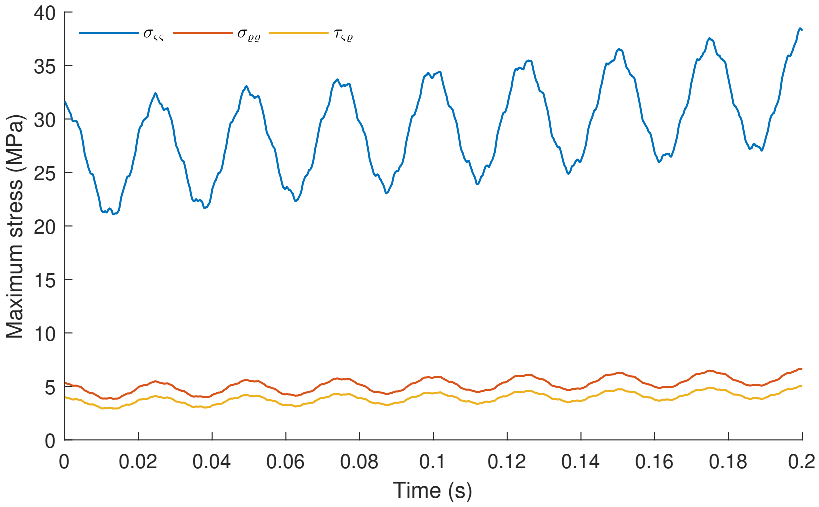


Figure 6.5: Temporal evolution of the maximum stresses (absolute value) for Ω_0 = 50 rad/s and α = 50 rad/s^2.

It is worth pointing that the mean value of $\sigma_{\varsigma\varsigma}$ increases with time, consequence of the increasing centrifugal force that is acting on the structure. Although only absolute values are represented in figure 6.5, looking into the dynamic results (figure D.21) one finds out that traction stresses are the most relevant contribution. As seen in the previous case, centrifugal force tends to pull the beam decreasing the effect of compression stresses.

Results animation

To ease visualisation of results, dynamic displacements and stress distribution will be animated in the rotation frame trough **GiD**. Only the case of higher angular acceleration (α = 50 rad/s^2) is presented in figure D.21. The animation captures one oscillation of the beam, with a timespan of 25 ms. As commented, higher stresses are found in the most deformed configurations of the beam.

6.2. Three-dimensional cantilever beam

For the next case of study, a new geometry will be defined. In order to keep the problem simple, a cantilever beam will again be the structure to study. In this case, however, the beam is modelled in three dimensions using 3200 hexahedral elements. The beam is 2 × 0.25 × 0.25 m, with a thickness of 0.05 m. Again, this study considers that the structure undergoes a rotation in the vertical -z-axis. The material from which the beam is made of is aluminium, with a density $\rho = 2700\, Kg/m^3$, a Young's modulus $E = 70\, GPa$ and a Poisson's coefficient $\nu = 0.3$.

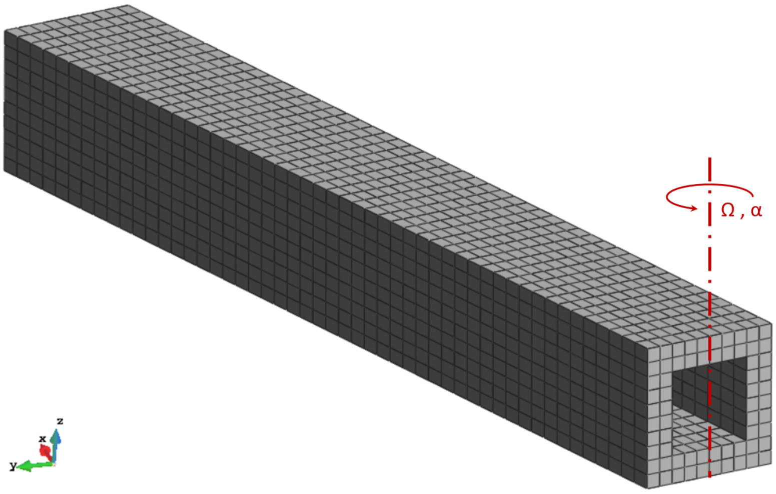


Figure 6.6: 3D cantilever beam subject to rotation around the vertical axis.

The loading scenario is the following: the beam is rotating at a given angular speed, subjected to $\alpha = 10$ rad/s^2, when the angular acceleration vanishes instantaneously and so does the tangential force. This section will analyse the vibrations this **sudden stop** in acceleration induces, for different values of angular speed Ω. Again, no source of damping will be considered. In contrast with the last analysis, this time the weight of the beam will be taken into consideration, and supposed constant and negative in the z axis. For comparison purposes, the first eight natural frequencies and modes of the presented structure can be found in D.2.1.

6.2.1. Constant angular velocity

For the sake of simplicity, the presented geometry will only be studied for the case of constant angular velocity. In fact, the dynamic behaviour will be analysed for three values of Ω: 25, 50 and 100 rad/s. The goal is to keep track of the vibration frequencies using the truncated SVD as Ω increases. For the numerical computations, the used timestep will be $\Delta t = 25\,\mu s$. To keep computation time low, the timespan has been selected in order to capture a full oscillation of first natural mode, so $T_f = 17\,ms$. For this case, the truncated SVD will compute the eight highest singular values and their associated LSVs and RSVs.

Campbell diagram

One of the main applications of the truncated SVD is to recover predominant frequencies of the dynamic vibration motion. The goal is to recover these frequencies for different values of Ω applying the discrete Fourier transform to each RSV. Recall that in the 2D case, predominant frequencies were compared with the softening effect predicted by modal analysis (see equation 5.25). However, this approach is not possible in three dimensions as the rotation stiffness matrix is no longer proportional to the mass matrix. The eigenvalue problem admits no further simplification than:

$$\left(\mathbf{K}_{\text{static}} + \mathbf{K}_{\text{rot}} - \boldsymbol{\omega}^2\,\mathbf{M}\right)\boldsymbol{\Phi} = 0 \tag{6.3}$$

The only way to keep track of the natural frequencies as angular velocity increases is by solving the previous equation for different values of $\mathbf{K}_{\text{rot}}(\Omega)$, obtaining **numerically** the desired Campbell diagram (ω $vs.$ Ω). By doing so, it is found that in most cases natural frequencies decrease in the same way as they did in 2D. The biggest **difference** is that in three dimensions, natural frequencies linked with modes acting in the z direction — the rotation axis — decrease much slower and in a different pattern.

This effect is displayed in a Campbell digram in D.29, where modal prediction and SVD results are compared. It is worth pointing that, in still conditions, natural modes and frequencies appear in **pairs**, as the beam is symmetric with respect to the $\varsigma\varrho$ and ςz planes. However, as rotation velocity increases, vibration frequencies of the modes acting on ϱ and z diverge, as their decrease rate is not the same (see Φ_7 vs Φ_8 in D.29).

Recall that one of the advantages of the SVD with respect to modal analysis is that it accounts for the effect of Coriolis, centrifugal and tangential forces. On the other hand, its main drawback is that as it is a **probabilistic method**, great timespans and small timesteps are needed to get a reliable value of predominant frequency. Nonetheless, the results displayed in D.29 agree ($\pm 1Hz$) with the predicted softening effect. It is found that the more intensively a mode is activated, the greater the accuracy in which the SVD can extract its predominant frequency. The RSVs from which the predominant frequency has been obtained are plotted for different values of Ω in D.24, D.26 and D.28.

Another limitation of the SVD is that it is not able to isolate natural modes. The LSVs the SVD computes are just predominant patterns, which quite often correspond to the combination of two or more natural modes. Figures D.23, D.25 and D.27 — which represent the predominant modes of the beam for different values of Ω — evince for instance how the first predominant mode $\mathbf{U}_1$ is in reality the combination of two natural modes: axial in ς ($\mathbf{\Phi}_6$) and bending in z ($\mathbf{\Phi}_2$). The higher Ω is, the higher contribution the axial mode has, as centrifugal forces increase with Ω^2 whereas weight is constant.

Displacements

Once more, displacements at the tip — upper side, at half the thickness — will be considered representative of the vibration phenomena. This time there is not a predominant direction in which displacements are considerably greater than in the others. Displacements in the ϱ axis are those with the lowest frequency and linked with the natural mode $\mathbf{\Phi}_1$ (see D.2.1), as the beam was under the effect of a **transverse force** which vanished at $t = 0$, oscillating thus around the equilibrium position $\varrho = 0$ (see figure D.30).

The structure undergoes as well a bending in the **vertical** direction z, but with a much higher frequency and a considerably smaller amplitude of oscillation. This effect is produced because, unlike in the ϱ direction, the distributed force in z — the weight — does not vanishes but remains constant as time passes by. Hence, vibrations in the vertical axis are not caused by the bending mode $\mathbf{\Phi}_2$ but are the result of higher frequency modes. Even though vibration amplitudes are relatively small in z, displacements on this axis have a considerable offset induced by the weight, as can be spotted in figure D.31.

A similar offset is found in the axial displacements, induced by the constant centrifugal force. As in the vertical direction, displacements in ς are subject to high frequency vibration, but to small oscillation amplitudes. One of the vibration frequencies corresponds to ω_1, as bending in ϱ also causes a small displacement in ς. For sure, there is as well a component linked with the axial mode $\mathbf{\Phi}_6$, but its amplitude is considerably smaller. These three effects — offset, ω_1 and ω_6 — can be easily identified In the following figure:

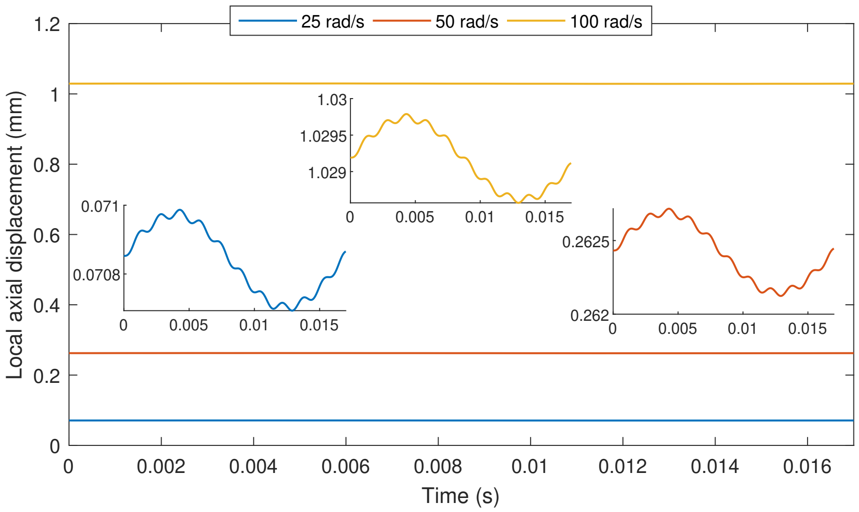


Figure 6.7: Temporal evolution of ς displacements at the tip of a cantilever beam under constant angular velocity.

With no surprise, all displacements **increase** (in absolute value) with Ω, being the ones in ς the more susceptible to changes in angular velocity, as the centrifugal force that acts in the axial direction is proportional to Ω^2.

Reactions

From a dynamic point of view, it is always interesting to keep track of the elastic reactions. Remember that like in the previous case, the beam is considered to rotate at a given angular speed, set by the driving device (recall the assumptions made in 2.1). This implies that no matter how great the deformations are, the device will always inject the necessary torque to ensure the desired kinematic conditions. Although it is an artificial scenario, it allows **uncoupling** of rigid body and elastic equations, simplifying considerably the dynamic study.

In this particular case, the elastic reaction torque in z oscillates with the frequency linked with the transverse motion, the most predominant one. The latter result is quite intuitive: the more deformed the configuration is, the more torque is required. When the deformation is in the direction of rotation, the reaction torque becomes negative. This oscillation in the torque is not caused by the effect of any real force, but because of the inertia induced by accelerations. On the other hand, the biggest component of the torque is actually constant, and produced by the centrifugal force. Thereby, the mean value of the elastic reaction torque in the rotation axis increases with Ω, as graphed in figure D.32.

Stresses

Once more, validation of the structure relies in the study of stresses. In order to simplify the analysis, only the maximum absolute value of stress has been dynamically tracked. Without considering any source of internal damping, these values are found to oscillate at twice the frequency of the flexural motion. Recall that stresses and strains adopt its higher value at maximum deformed configuration, which takes place twice per period. With all, it has been found that like in the 2D case, the higher stresses are those in the direction of the centrifugal force — that is, $\sigma_{\varsigma\varsigma}$. Maximum stresses for $\Omega = 50$ rad/s are represented in the following figure:

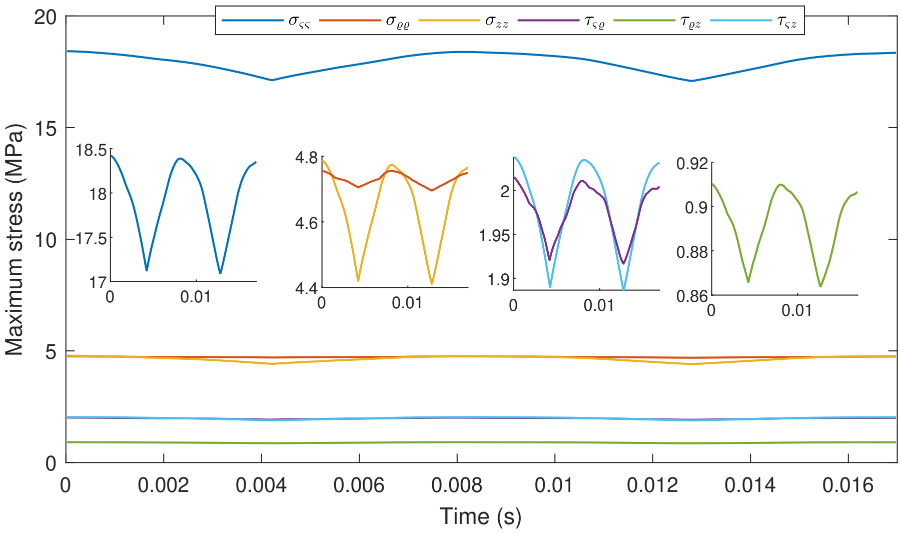


Figure 6.8: Temporal evolution of the maximum stresses (absolute value) for $\Omega = 50$ rad/s.

In 3D cases where six different components of stress play a significant role, its worth considering the equivalent stress. Using the **Von Mises** yield criterion for beams and considering a 6061 T6 alloy, it is found that the security factor of the structure is 14.5.

Results animation

The dynamic results have been postprocessed and animated using **GiD**. The deformed configuration and the stress map for different timesteps are presented in figure D.33. For the sake of simplicity, only $\sigma_{\varsigma\varsigma}$ stresses are coloured. Regarding displacements, it is relatively difficult to keep track of the bending motion in ϱ, as axial displacements are considerably high and **eclipse** the flexural vibration.

7

Study of an aeronautics-related case

So far, the presented chapters encompassed a relatively abstract subject, and has only been put in practice to test the capabilities of the SVD in beam analysis. The proposed cases where artificial scenarios that could be easily studied by following the hypotheses posed in 2.1. The aim of this chapter is to modify the developed numerical tool in order to tackle a case of aeronautical interest, which is the starting of an **helicopter rotor**.

7.1. Case of study

The main goal of the presented chapter is the study of the dynamic behaviour of an helicopter's rotor during the starting of the engines. That is, the blades will start from idle configuration and will end rotating at its **design** angular velocity. Helicopter dynamics is a wide and complex subject combining both structural analysis and aerodynamics. As the understanding of the whole phenomena is not the goal of the present report, the following hypotheses will be assumed:

- Blades will be modelled as two-dimensional structures. Three-dimensional effects (structural and aerodynamic) will be neglected, as well as gravity forces.
- Rotation axis is fixed and perpendicular to the plane of the blade.
- The fluid will not be simulated but rather aerodynamic forces will be modelled.
- The rotor will produce zero lift, and thus, no induced velocity along the span of the beam will be generated. Parasite drag is the only aerodynamic force acting on the rotor.
- The helicopter is performing hovering flight with no wind speed.
- The blade is fixed-free: no articulated parts are considered. Hence, the mechanical twist angle will always be zero.

The rotor is made of three equally distributed blades, whose geometry is defined in figure 7.1. The radius of the blade is $R = 5\,m$, whereas the chord is 0.2 and 0.4 m at the root and the tip, respectively. The maximum chord is 0.55 m and its located at the 8% of the span. The total surface of the blade is $A_{\text{blade}} = 2.335\,m^2$. The blade has been modelled using a CAD software (***SolidWorks***) and then exported to **GiD** to proceed with the mesh generation. Its geometry is defined in figure 7.1.

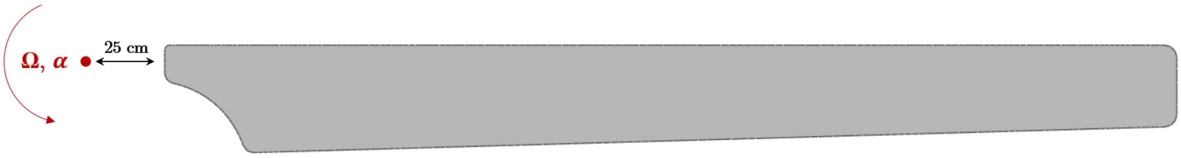


Figure 7.1: Geometry and of a 2D blade.

In order to mesh the above geometry using only quadrilateral elements, the blade needs to be modified. By eliminating curves and roundings, the blade is discretised using a structured mesh made of 1340 elements:

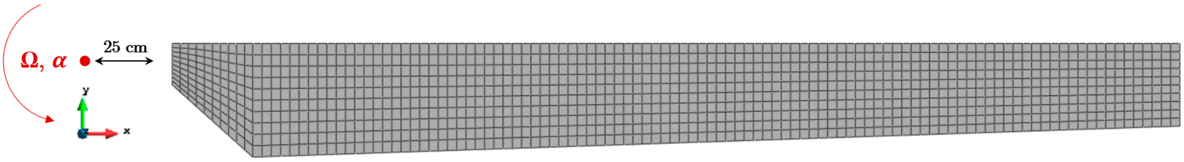


Figure 7.2: Structured mesh of a simplified 2D blade.

In three dimensions, the blades can be thought as an extrusion of a *NACA 0012* airfoil, whose cross section $A_{\text{airfoil}} = 0.02081\, m^2$ has been computed integrating the profile shape [15]. To approximate the superficial density of the blade, it will be considered that its total mass is one fifth of the mass of a solid aluminium blade:

$$m = \frac{1}{5}\rho\,(A_{\text{airfoil}} \cdot R) \approx 56.19Kg \;\Rightarrow\; \rho_s = \frac{m}{A_{\text{blade}}} \approx 24\,\frac{Kg}{m^2} \tag{7.1}$$

Regarding the **driving device**, it will be supposed that the helicopter has an alternative engine able to inject power to the rotor. As engines cannot produce power when the angular velocity is zero, an electric motor will be used to generate a torque that starts the rotation motion. Remember that the power P of a rotating object is given by:

$$\mathrm{P} = \Omega\,\mathrm{T} \tag{7.2}$$

Where T is the torque produced by the device. To avoid sudden jumps in the variables, neither the torque nor the power will be injected instantaneously. Instead, they will vary from an initial to a final value describing a **sinusoidal** shape:

$$\begin{gathered}
\mathrm{T} = \mathrm{T}_0 + \mathrm{T}_{\text{nom}} \sin\left(\frac{\pi}{2\mathrm{t}_a}\,\mathrm{t}\right) \qquad 0 \le t < \mathrm{t}_a\,, \qquad\qquad \mathrm{T} = \mathrm{T}_{\text{nom}} \qquad \mathrm{t}_a \le t < \mathrm{t}_b \\
\mathrm{P} = \tfrac{1}{2}\left(\mathrm{P}_0 + \mathrm{P}_{\text{nom}}\right) + \tfrac{1}{2}\left(\mathrm{P}_{\text{nom}} - \mathrm{P}_0\right) \sin\left(\frac{\pi}{\mathrm{t}_c - \mathrm{t}_b}\,\mathrm{t} + \pi\left(\frac{1}{2} - \frac{\mathrm{t}_c}{\mathrm{t}_c - \mathrm{t}_b}\right)\right) \qquad \mathrm{t}_b \le t < \mathrm{t}_c \\
\mathrm{P} = \mathrm{P}_{\text{nom}} \qquad\qquad \mathrm{t}_c \le t
\end{gathered} \tag{7.3}$$

W_0 and T_{nom} are the initial and nominal values, respectively, of torque generated by the electric motor. The motor reaches the nominal value of torque at t_a, and maintains it constant until t_b, when the engine starts operating. To avoid discontinuities in the injected power, the following statement has to be met:

$$\mathrm{P}_0 = \Omega_{(\mathrm{t}_b)}\,\mathrm{T}_{\text{nom}} \tag{7.4}$$

The engine starts to operate at t_b and reaches its design value P_{nom} at t_c, maintaining this value constant for further values of time until normal flight conditions are reached (see figure 7.3). No automatic control that regulates the power as a function of the angular velocity is taken into consideration.

7.2. Rigid-elastic coupling

Unlike in previous chapters, kinematics of the rotating motion are no longer known. This implies breaking the first hypothesis defined in 2.1, and thus, the developed model is no longer applicable. In previous cases (see 6), angular velocity and acceleration were known at every time, being the reaction forces the unknowns. In this case, what is known is the power provided by the engine, whereas the unknowns are the angular variables θ, Ω and α. This is a major change in the problem formulation, as rotation and vibration motions become **coupled**. From solid mechanics, it is found that the relationship between the torque and the angular velocity of a rigid body is given by:

$$\mathbf{W}_a = \frac{\mathrm{d}\,(\mathbf{J}_a\,\boldsymbol{\Omega})}{\mathrm{dt}} + \mathbf{AG} \times \mathrm{m}\,\mathbf{a}_a \tag{7.5}$$

Where a is a generic point where the torque is to be evaluated, $\mathbf{J}_a$ is the inertia tensor of the solid with respect to a, $\boldsymbol{\Omega}$ is the vector of angular velocities, $\mathbf{a}_a$ the linear acceleration of the point a and **AG** the vector pointing from a to the centre of mass of the solid. If the inertia matrix is posed in such a way that is **constant** in time, the above expression transforms into Euler's equations for rigid bodies (see 1.2). Luckily enough, with the assumed hypotheses, the torque equation can be expressed as stated in 3.22:

$$\mathrm{W}_z = \mathrm{J}_z\,\alpha \tag{7.6}$$

And taking into account that W_z results from the contribution of the torque generated by the engine and the elastic reaction torque induced by the structure:

$$\mathrm{T} - \mathrm{W}_{el} = \mathrm{J}_z\,\alpha \tag{7.7}$$

And recall how the elastic reaction torque was computed:

$$\mathrm{W}_{el\,(i)} = \sum_{j=1}^{n_{pt}} \mathbf{w}_{el\,(i\cdot j)} = \sum_{j=1}^{n_{pt}} \left(\begin{bmatrix} \Delta\widetilde{\mathbf{r}} \\ \vdots \end{bmatrix} \mathbf{R} \right)_{(i\cdot j)} \tag{7.8}$$

Accounting only for the torque in the rotation axis, and including equation 3.21 equation 7.2, we arrive at:

$$\mathbf{M_{ll}\ddot{d}_l} + \mathbf{D_{ll}\dot{d}_l} + \mathbf{K_{ll}d_l} = \mathbf{F_l} - (\mathbf{M_{lr}\ddot{\overline{u}}} + \mathbf{D_{lr}\dot{\overline{u}}} + \mathbf{K_{lr}\overline{u}})$$

$$\mathrm{J}_z\,\alpha = \tfrac{\mathrm{P}}{\Omega} - \textstyle\sum_{j=1}^{n_{pt}} \left(\begin{bmatrix} \Delta\widetilde{\mathbf{r}} \\ \vdots \end{bmatrix} \left[\mathbf{M_{rl}\ddot{d}_l} + \mathbf{D_{rl}\dot{d}_l} + \mathbf{K_{rl}d_l} + \mathbf{M_{rr}\ddot{\overline{u}}} + \mathbf{D_{rr}\dot{\overline{u}}} + \mathbf{K_{rr}\overline{u}} - \mathbf{F_r}\right] \right)_{(3\cdot j)} \tag{7.9}$$

The main obstacle in the resolution of the above system is that all **K**, **D** and **F** depend on both Ω and α. Moreover, the way in which the previous matrices are defined (see 3.13) make the equations highly **nonlinear** in Ω and α. Hence, a proper numerical method has to be proposed in order to tackle the presented system of non-linear differential equations.

7.3. Numeric integration

In the context of coupled rigid-elastic dynamics, the unknowns are the vectors $\mathbf{d}$, $\dot{\mathbf{d}}$ and $\ddot{\mathbf{d}}$ regarding the vibration motion, and the scalars θ, $\dot{\theta}$ and $\ddot{\theta}$ regarding the rotation motion. Using the presented **Newmark** $\boldsymbol{\beta}$**-method**, the previous system of differential equation can be transformed into a system of algebraic equations. The transformation this method proposes has already been reviewed in 5.2 for the elastic variables, and the only thing left is to apply it to the angular variables:

$$\Omega^{n+1} = \Omega^{n} + (1-\gamma)\Delta\mathrm{t}\,\alpha^{n} + \gamma\,\Delta\mathrm{t}\,\alpha^{n+1} \tag{7.10}$$

Hence, what is left is a system of non-linear **algebraic** equations with the unknowns α^{n+1} and $\mathbf{d}^{n+1}$. This problem could be tackled through a non-linear solving algorithm such as Newton-Raphson, widely used in computational mechanics. It is an iterative procedure based on the solving of a sequence of linear models [12]. The Newton method can in fact be applied to arbitrary systems of algebraic equations with n unknowns. However, its implementation is not an easy task and requires a great understanding of algebra. Moreover, as it is an iterative algorithm, it requires the different matrices to be computed more than once at each timestep, increasing considerably the computation time.

A simpler approach is based on the **explicit** central difference scheme, which computes accelerations at t^n and uses this value to update velocities and displacements at t^{n+1}. Although very easy to program, explicit methods require from very **little** timesteps to ensure conditional stability. Thus, computation time will arise with respect to the previous case of known kinematics. The general algorithm can be summarised as:

1. Obtain $\mathbf{d}^{n+1}$ solving the system of equations stated in 5.34 (using Ω^n and α^n) .
2. Once displacements at t^{n+1} are known, compute the elastic reaction torque W_{el}^{n+1} from 7.9.
3. Find α^{n+1} from equation 7.7 and update Ω^{n+1} using expression 7.10.

The initial value of angular accelerations is found by solving the non-linear algebraic equation that involves initial conditions $\mathbf{d}_0$, $\dot{\mathbf{d}}_0$, $\ddot{\mathbf{d}}_0$ and Ω_0. As α_0 is the only unknown, the equation is solved by using a simple **fixed-point iteration**, using as initial guess $\alpha_0 = 0$. It has been found that four iterations suffice to obtain a converged solution.

7.4. Aerodynamic forces

Performing a simulation that accounts for fluid-structure interaction is far out of the scope of this section. Instead, the aerodynamic forces acting on the blade will be **modelled** using simplified aerodynamic relations. As the rotor is modelled in two dimensions, no forces acting perpendicular to the plane will be taken into consideration. Following the hypotheses presented in 7.1, the only force acting on the rotor plane is the parasite drag.

This aerodynamic resistance force depends on many parameters, such as the angle of attack, the surface roughness of the blade, the Mach number and the Reynolds number — a non-dimensional parameter that measures the strength of inertial forces of the fluid with respect to viscous ones. In order simplify things up, the effect of the latter variables will be encapsulated in a non-dimensional **drag coefficient**, c_D, which will be considered known. Otherwise, this analysis would be impracticable. The drag differential (dD) acting on a surface differential (dA) of the blade is given by:

$$\frac{\mathrm{dD}}{\mathrm{dA}} = \frac{1}{2}\,\rho_{\,\mathrm{air}}\, c_D\, \mathrm{v}^2 \tag{7.11}$$

Where v stands for the relative velocity of the surface differential with respect the fluid. As the blades are rotating in a plane, their solid rigid velocity is proportional to $\Omega\,\mathbf{r}_i$, where $\mathbf{r}_i$ stands for the initial position. Taking into account the velocity induced by the elastic vibrations, and the change in radial position due to elastic displacements:

$$\mathbf{v}^2 = (\tilde{\Omega}\,(\mathbf{r}_i + \mathbf{u}) + \dot{\mathbf{u}})^2 = \left[\dot{\mathbf{u}}^2 + 2\,(\tilde{\Omega}\,\mathbf{u})\,\dot{\mathbf{u}} + 2\,(\tilde{\Omega}\,\mathbf{r}_i)\,\dot{\mathbf{u}} + (\tilde{\Omega}\,\mathbf{u})^2 + 2\,(\tilde{\Omega}\,\mathbf{r}_i)\,(\tilde{\Omega}\,\mathbf{u}) + (\tilde{\Omega}\,\mathbf{r}_i)^2\right] \tag{7.12}$$

As the geometry is 2D, the drag force can be modelled in terms of the FEM as a **body force**, proportional to the area of each element. In three-dimensional problems this approach would not be correct, as aerodynamic forces only act on the surface and hence are to be modelled as boundary tractions. With this in mind, the term dD is to be inserted as a body force $\boldsymbol{f}$ in equation 3.13. As a result, the following expression is obtained:

$$\boldsymbol{f}_D = \int_{\mho} \mathbf{N}^T \frac{1}{2} \rho_{\text{air}} c_D \mathbf{v}^2 \, \mathrm{d}\mho \tag{7.13}$$

The problem is that from $\boldsymbol{f}_D$ will not only emerge components proportional to $\dot{\mathbf{u}}$ and $\mathbf{u}$, but also non-linear terms such as $\dot{\mathbf{u}}^2$, $\mathbf{u}^2$ and $\dot{\mathbf{u}}\mathbf{u}$. The responsible for these nonlinearities to appear is the term $\mathbf{V}^2$ defined in 7.12. To make the problem more attainable, it will be supposed that $\widetilde{\Omega}\,\mathbf{r}_i \gg \dot{\mathbf{u}}$ and $\mathbf{r}_i \gg \mathbf{u}$. Moreover, air density and drag coefficient will be considered constant along the domain. These hypotheses allow $\boldsymbol{f}_D$ to be simplified as:

$$\boldsymbol{f}_D = \frac{1}{2} \rho_{\text{air}} c_D \int_{\mho} \mathbf{N}^T \, \widetilde{\Omega^2} \, \mathbf{r}_i^2 \, \mathrm{d}\mho \tag{7.14}$$

The above integral is decomposed into elemental matrices as reviewed in B, and computed using Gauss Quadrature. The major drawback of this simplified scenario is that drag forces no longer account for the effect of vibration, and thus air does not damp oscillating motions. In order to overcome this limitation, internal Rayleigh **damping** will be taken into consideration (recall equation 3.15). This cushioning will induce a new sources of stresses as expected from equation 2.15:

$$\boldsymbol{\sigma} = \mathbf{C}\,\boldsymbol{\varepsilon} + \overline{\mathbf{D}}\,\dot{\boldsymbol{\varepsilon}} \;\rightarrow\; \mathbf{C}\,\nabla^s \mathbf{N}\,\mathbf{d} + \overline{\beta}\,\mathbf{C}\,\nabla^s \mathbf{N}\,\dot{\mathbf{d}} = \mathbf{C}\,\mathbf{B}\,\mathbf{d} + \overline{\beta}\,\mathbf{C}\,\mathbf{B}\,\dot{\mathbf{d}} \tag{7.15}$$

To close down this section, let's make an estimation of how much **power** is required to compensate the parasite drag when operating at nominal speed, say 400 rpm. Considering a constant chord c, the power the aerodynamic resistance of each blade induces is:

$$\mathrm{P}_{\text{drag}} = \Omega \int_S \varsigma \, \mathrm{d}\mathrm{D} = \Omega \int_0^R \frac{1}{2} \rho_{\text{air}} c_D \mathbf{v}^2 \, c \, \varsigma \, \mathrm{d}\varsigma = \frac{1}{2} \rho_{\text{air}} c_D \Omega^3 \, c \int_0^R \varsigma^3 \, \mathrm{d}\varsigma = \frac{1}{8} \rho_{\text{air}} c_D \Omega^3 \, c \, R^4 \tag{7.16}$$

Considering flight at sea level ($\rho_{\text{air}} = 1.225\,Kg/m^3$), a mean chord $c = 0.45\,m$ and a drag coefficient $c_D = 0.008$, each blade requires from $25.32\,kW$ to surpass aerodynamic resistance. Nevertheless, the engine will need to inject more power to counteract the elastic reaction torque.

7.5. Rotor starting simulation

Once the FEM software is updated with the developed models from the previous sections, it is time to test it out simulating the starting of a rotor blade. Imagine that the studied helicopter has the following properties:

- **Number of blades**: 3
- **Mean chord**: 0.45
- **Nominal speed**: 400 rpm
- **Motor nominal torque**: 180 kN · m
- **Engine nominal power**: 580 kW
- **Nominal torque given at**: t_a = 25 ms
- **Engine starting time**: t_b = 50 ms
- **Nominal power given at**: t_c = 120 ms

From basic helicopter theory, it is found that for hovering flight, a simple approximation of the power required to compensate the weight is:

$$\mathrm{P}_{\text{lift}} = \frac{(m\,g)^{\frac{3}{2}}}{\sqrt{2\,\rho_{air}\,\pi\,R^2}} \tag{7.17}$$

Hence, considering an helicopter with a mass $m = 2800\ Kg$, the power required to ensure hovering conditions at sea level is $\mathrm{P}_{\text{lift}} = \ 328.18\,kW$. Subtracting this value to the nominal power, it is found that the **available power** to counteract the elastic reaction torque is around 84 kW per blade, which is higher than the parasite power induced by aerodynamic drag obtained in equation 7.16.

To be conservative, it will be supposed that the engine is not operating at nominal power but at its 60% (50 kW per blade). Regarding the electric motor, it will be considered that at the beginning of the rotation motion the blades are undeformed, and hence all the torque — 60 kN · m per blade — is used to **accelerate** the rotor. For sure, this development is an oversimplification of the real scenario, but still is good enough to provide a glimpse on how the blade dynamically behaves.

As the numerical integration requires from very small timesteps ($\Delta t = 5\mu s$), the timespan is limited to 5 s. To reduce computation time, stresses will only be computed every 20 timesteps. Dynamic displacements will also be saved at the same rate to reduce memory consumption. The integration is going to be performed using the constant average acceleration scheme. The considered structural Rayleigh damping is given by $\overline{\alpha} = 0.25$ and $\overline{\beta} = 5 \cdot 10^{-6}$. With these inputs, the simulation takes up to three hours to complete.

The full results of the simulations are presented in the appendix (section D.3). The temporal evolution of the presented variables is **closely linked** with the way torque is injected into the blade, as angular accelerations are proportional to it. Different input torque will result in a completely different dynamic behaviour of the blade. For the studied case, torque and power are displayed in the following figure:

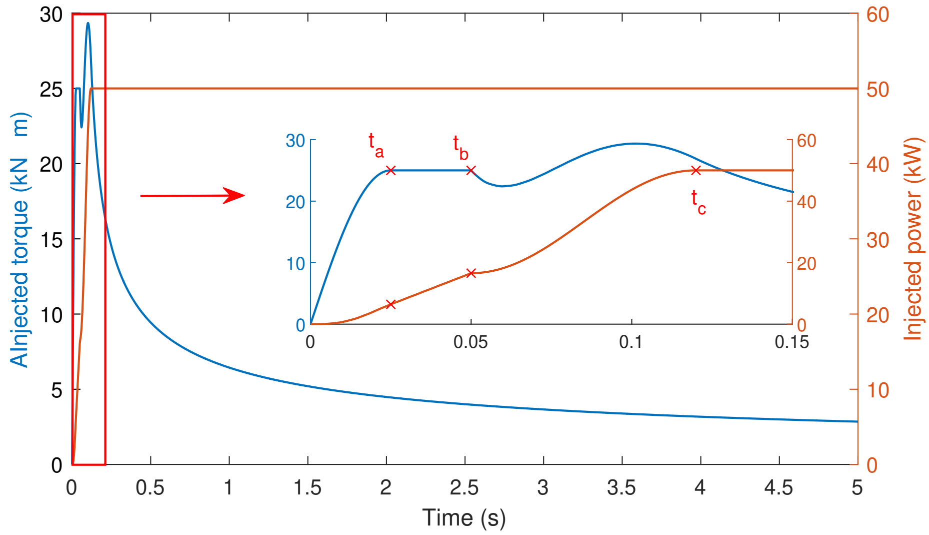


Figure 7.3: Injected power and torque as a function of time.

Considering displacements at the tip of the blade as representative from the vibration motion, it is found that the blade is not undergoing continuous oscillation around and equilibrium configuration. Although it is true that oscillations are governing at the very beginning, displacements in ϱ tend to reach a non vibrating **steady state**. This effect was not sighted in previous simulations, where vibrations persisted in time. In all likelihood, the reduction of high frequency vibrations is induced by the structural damping acting on the blade. In fact, the SVD has been able to recover predominant **damped oscillations** — see RSVs 3 and 4 in figure D.37 — that would otherwise be impossible to spot.

Much the same can be said about local axial displacements, but this time their tendency is not towards equilibrium but into a raise in time. This same behaviour has been spotted in earlier simulations, as the centrifugal force inducing axial displacements **increases** with Ω^2. Both displacements in ς and angular speed are compared in figure D.34.

Regarding the forces acting on the blade, there is a considerable *peak* at the beginning of the rotation motion, when the motor is injecting torque to the standstill rotor. During these first instants, all forces, accelerations and even stresses undergo oscillations. As time passes by and the engine achieves its nominal conditions, both reaction forces and accelerations **stabilise** into a considerably lower value. The only exception has to do with the torque induced by aerodynamic drag, that increases with the angular speed. In figure D.35, angular acceleration is graphed along with the reaction torque, showing how acceleration increases when reactions decrease.

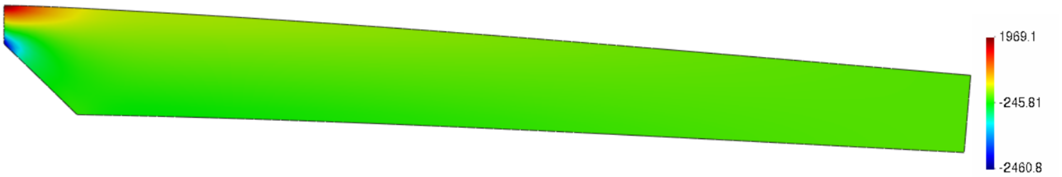


Figure 7.4: Deformed configuration (×1000) and $\sigma_{\varsigma\varsigma}$ (KPa) of the blade for the moment of maximum stress.

When reviewing maximum stresses, it is found that the peaks correspond to the maximum deformed configuration. Again, this value reaches a steady condition as time passes by. What it has been found when carefully checking the results is that, after a few seconds, stresses tend to raise again. This effect is caused by the increase of aerodynamic drag, that induces forces proportional to Ω^2. This same condition is found in transversal displacements, hinting that there is a certain value of angular speed at which stresses and deformations are **minimum**. This effect can be spotted in the following figure, where these minimums are represented with a red cross:

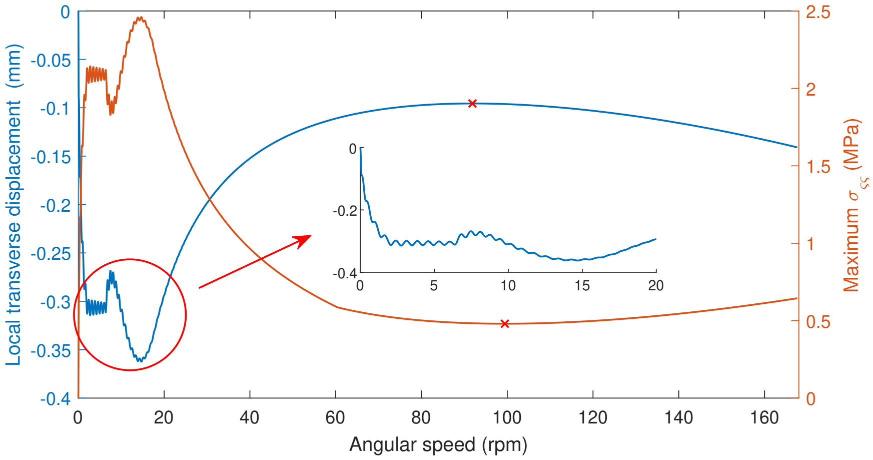


Figure 7.5: Temporal evolution of ϱ displacements at the tip of the beam and maximum $\sigma_{\varsigma\varsigma}$.

It is worth pointing that in the simulated timespan, the angular velocity of the rotor has not reached its nominal value, but further timespans are far too restrictive in terms of computation time. The rotor would eventually surpass the nominal speed, as no **control mechanism** has been considered. To overcome this issue, injected power needs to be reduced when approaching the desired conditions.

8 Closure

In this final chapter the overall budget of the study is examined, the developed work is reviewed and final conclusions are stated. Possible future tracks to continue with the present study are also discussed.

8.1. Budget

Since the major part of the work developed during this thesis is focused on studies and analyses, the greatest expense is related with the **salary** of the researcher, which has been estimated to 9560 € for 478 working hours. Accounting for the equipment cost — essentially software licenses and basic hardware — the overall budget of the present project raises up to **13450** €. The reader may consider consulting the attached document *'Budget'* to get a better clue on the costs analysis.

8.2. Conclusions

Rotordynamics has been a field of great interest for centuries, being credited for the huge improvement in rotating machinery since the industrial revolution. Modern rotating components such as turbines, rotor blades and engines are still being designed using classical models developed centuries ago. What really created a game-changing shift in rotordynamics was the emergence of **numerical methods** in the earlier sixties. Implementation of the FEM gave rise to a completely new approach for structural analysis, allowing complex geometries to be solved in very short amounts of time.

Nowadays, the use of numerical software in structural analysis is a quite common solution. Despite of the great improvement in numerical capabilities over the last years, the FEM still requires from great processing power to perform complex simulations. Hence, it is not surprising that the actual tendency goes in the direction of more efficient methods instead of more accurate models. In this context, the present report has reviewed some of the available **numerical tools** to tackle the equations describing the vibration behaviour of a rotatory blade.

This thesis started with the elastic modelling of a simplified rotor blade undergoing one great rotation. The model supposes small strains and neglects the gyroscopic effect. When the obtained equations are posed in terms of the FEM, a **softening effect** is unfolded. The latter effect causes natural frequencies to decrease with rotation speed. However, real rotors do not exhibit this behaviour, but instead are affected by an opposite stiffening effect. Hence, the simplified model is not capable to capture the complete vibration phenomena, but even so it is a great tool that offers a glimpse on how rotordynamics solvers work.

For the static case, the developed FEM tool has proven to provide **correct results**, as it has been validated with analytic expressions. Regarding the dynamic scenario, the present work has focused on the numeric methods itself rather than on the description of the physical model. The goal of this thesis is not to bring to light new discoveries in the field of rotordynamics, but to give a glimpse on how numeric schemes work and in which way could they be more performing. The problem with structural analysis is that the governing equations rapidly become **nonlinear**, being numerical integration the only possible path to handle them.

For that purpose, a numerical scheme based on the Newmark-β method has been developed to study the dynamic response of rotating structures. It has been found that the rotation condition increases considerably the computational costs of the simulations, as matrices linked with rotating effects need to be computed and assembled at each timestep. In order to reduce simulation times, a more **efficient assembly** method has been used instead of the traditional approach, leading to an increase in performance by a factor of 15.

On the opposite side, when the physics describing the dynamic behaviour of a system are simple, **modal decomposition analysis** can be used to study its vibration behaviour. Modal analysis is a widespread tool that unfolds relevant information about the system with low computation costs. In structural analysis, modal decomposition is used to recover natural frequencies and mode shapes. The biggest drawback of this method is its inherent limitation to work only with simple dynamic systems.

Being aware of this situation, the truncated SVD has been found to be an alternative method that combines both numerical integration and modal decomposition to recover predominant patterns of the dynamic behaviour of the structure. If applied to structural analysis, it has been proven to recover predominant modes and frequencies of the system with **great accuracy**. The studied cases implied 2D and 3D rotating cantilever beams, although it could be extended to any arbitrary geometry. In order to recover oscillation frequencies from predominant patters, the developed method uses the **discrete Fourier transform** to catch the higher components in the frequency domain. One has, however, to keep in mind that the SVD is a probabilistic method, and thus gains reliability as the size of the sample increases.

The reader could come to think that the SVD entails no real advantage, as the input data matrix is still computed using numeric integration. Nothing further from the truth: the proposed method allows a **deeper understanding** of the vibration phenomena as it breaks up the whole problem into simpler components, easier to interpret and to postprocess. The postprocessing becomes thereby much simpler and faster, as the complete behaviour of the structure can be decomposed into a linear combination of **predominant** modes and frequencies. Hence, the problem numerical integration has about encountering multiple time and space scales in the postprocessing step is overcome. This make results analysis an easier and quicker task for the engineer, reducing both the time and computation resources consumed.

When testing the SVD in rotating beams, it has been found that the method is able to keep track of the softening effect predicted by the model. This capacity of capturing the reduction in vibration frequencies as angular velocity increases would be unthinkable through traditional approaches.

Moreover, it has been spotted that the dynamic displacements can be approximated using only a few singular values, allowing an extensive reduction in postprocessing requirements.

In order to locate the developed work in the context of **aerospace** engineering, the built software has been adapted to work together with a rigid body model. This has allowed the transient study of the starting of an helicopter's rotor, giving plausible results and proving the actual capabilities of the method.

When thinking about the potential capabilities of the SVD in structural analysis, it is too soon to reach a conclusion. The work is still in a very **early state**, and much investigation needs to be carried in this subject. This report has only accounted for a very simple rotor model, so nothing can be said about the validity of the results as there is no way they could be compared with empirical data. Until the proposed method is further developed, the present thesis will remain as a theoretical but plausible approach for real rotordynamics studies.

8.3. Future development

The present report has reached nearly all the defined objectives. However, as a result of the short amount of time available to carry out the current work, some aspects still admit further improvement.

Model refinement

A first possible upturn has to do with the modelling of the rotor as an elastic solid. In rotordynamics, structures not only undergo rotations in the axis imposed by the driving device but also oscillate in a whirling motion. This combination of rotations induces the **gyroscopic effect** to emerge. The latter condition, although defined by relatively simple rigid body equations, hinders the analysis of rotating structures as their moment of inertia and the way it changes in time plays a non-negligible role in the dynamic response.

When the rotation axis changes in time, analysis in the corotational frame becomes more complicated. For that reason, it would be interesting to study the vibration phenomena of rotating structures in the inertial frame. In the most general case, the structure undergoes great rotations that cannot longer be tackled with the presented method. Other methodologies such as **domain decomposition** could be a great approach in handling structures under the effect of great rotations in the small strains range.

The last improvement of the model has to do with the restriction of small strains, as the forces induced in rotating machinery can produce great displacements which is not possible to be modelled using the linear elastic model. With those new considerations, this new model should be able to predict with considerable accuracy the **stiffening effect** that the presented method has not been capable to capture.

Real case application

One of the strongest limitations of the presented work is the restriction to work only with simple geometries, given the humble computation resources available. Moreover, definition and meshing of complex structures is not a piece of cake, and requires from further experience in the field. If a real scenario could be modelled, accounting for instance for bearings and damping sources, then the numeric results could be compared with **experimental data**. The validation of every numeric code is an essential step in simulation of physic phenomena, which unfortunately has not been accomplished in the present report.

Fluid–structure interaction

Little has been done in the present work regarding the processing of the surrounding fluid. In chapter 7, aerodynamic drag was taken into consideration using simple models, but real scenarios require from more precise simulations to introduce, for example, the effect of the 3D wake generated by the rotor. Fluid dynamic simulations were far out of the scope of the current report, as the fluid constitutive equations are highly nonlinear and one of the hardest to solve without high performance CFD software.

The ideal goal of a coupled fluid-structure software would be to capture their **mutual interaction**. The fluid induces a pressure distribution along the surface of the structure, which is responsible of the aerodynamic forces that produce deformations in the solid.These deformations produce perturbations in the flow, which results in a change in the aerodynamic forces. If this phenomena is fully understood, then fields of great interest such as flutter could be predicted.

Code optimisation

During the entire development of the present work, all the software has been coded in MATLAB. Although it is by far the most convenient choice in an educational project, when it comes to performance it has nothing to do with other programming languages such as *C* or *Fortran*. Hence, great improvements in computation terms could be achieved by properly rewriting the code into a more performing language.

On the other hand, postprocessing with **GiD** is slightly inefficient, as it uses ASCII code to read and display the dynamic results, which results in slow importations and heavy animations. The idea of using a more convenient software to perform both pre-and postprocessing should be taken into consideration.

SVD enhancement

The SVD has been proven to be an extremely powerful tool to tackle structural problems. It allows extraction of predominant modes and frequencies in scenarios where modal analysis is no longer applicable. However, the proposed method of extracting predominant frequencies using the discrete Fourier transform to map the RSVs into the frequency domain should be revised, as the used in-built MATLAB functions may not be the best approach.

In the course of this report, the SVD has only been used to obtain truncated models of displacements, catching statistically predominant patters of the dynamic response. However, the SVD could theoretically be applied to **further variables**, being able to recognise predominant stress distributions and even force patters typical from resonance conditions. If combined with modern numerical tools such as artificial intelligence and machine learning, the SVD could be systematically used to keep track of dynamic systems and even make predictions of potential scenarios using low computation resources.

Bibliography

[1] J. S. Rao, *History of rotating machinery dynamics.* Springer, 2011.

[2] G. Genta, *Dynamics of Rotating Systems.* Springer-Verlag New York, 2005.

[3] G. Genta and M. Silvagni, "On Centrifugal Softening in Finite Element Method Rotordynamics," *Journal of Applied Mechanics*, vol. 81, 2013.

[4] J. R. Bentley, "Aeolipile - Hero's Ball: A Replica of the World's First Rotating Steam Engine." `http://modelengines.info/aeolipile/`. Accessed: 2010-04-16.

[5] D. Volkov, "Heinkel He.178." `http://www.airwar.ru/enc/xplane/he178.html`. Accessed: 2019-04-16.

[6] H. Gao and G. Bai, "Reliability analysis on resonance for low-pressure compressor rotor blade," *Advances in Mechanical Engineering*, 2015.

[7] D. Roylance, "Beam Displacements. Department of Materials Science and Engineering, Massachusetts Institute of Technology," `https://ocw.mit.edu/courses/materials-science-and-engineering/3-11-mechanics-of-materials-fall-1999/modules/MIT3_11F99_bdisp.pdf`. Accessed: 2019-04-12.

[8] J. A. González, R. Abascal, and K. C. Park, "Partitioned analysis of flexible multibody systems using filtered linear finite element deformational modes," *International Journal for Numeric Methods in Engineering*, pp. 102–128, 2014.

[9] UPC ETSEIB, "Numerical Factory - Finite Elements: a first course." `https://numfactory.upc.edu/web/FiniteElements.html`. Accessed: 2019-05-08.

[10] Tulloch Andrew, "Fast Randomized SVD - Facebook Research." `https://research.fb.com/fast-randomized-svd/`. Accessed: 2019-05-16.

[11] D. J. Inman, *Engineering Vibration.* Pearson, 2014.

[12] T. Belytschko, W. K. Lin, B. Moran, and J. Wiley, *Nonlinear finite elements for continua and structures.* John Wiley & Sons, 2000.

[13] J. Hernández, J. Oliver, and A. Huespe, *Computational Homogenization of Inelastic Materials using Model Order Reduction.* 2014.

[14] W. D. Callister and D. Rethwisch, *Materials Science and Engineering: An Introduction.* John Wiley and Sons, 1985. 8th edition.

[15] "Airfoiltools - Airofil Database." `http://airfoiltools.com`. Accessed: 2019-06-03.

[16] O. Matsushita, M. Tanaka, H. Kanki, M. Kobayashi, and P. Keogh, *Vibrations of Rotating Machinery - Basic Rotordynamics: Introduction to Practical Vibration Analysis*, vol. 1. 2017.

[17] M. Géradin and A. Cardona, *Flexible Multibody Dynamics - A finite elements approach.* John Wiley & Sons, 2001.

[18] C. Japhet, F. Cuvelier, and G. Scarella, "An efficient way to perform the assembly of finite element matrices - <hal-00931066v2>," 2015.

[19] A. A. Shabana, "Flexible Multibody Dynamics: Review of Past and Recent Developments," *Multibody System Dynamics*, pp. 189–222, 1997.

[20] N. I. GUWAHATI, "Theory of Rotor Dynamics." `https://nptel.ac.in/courses/112103024/23`. Accessed: 2019-05-21.

[21] D. V. Hutton, *Fundamentals of Finite Element Analysis.* Mc Graw Hill, 2004.