

Developing a Digital Twin for the Dynamic Analysis of Jacket Foundations for Offshore Wind Turbines

Udvikling af en digital tvilling for dynamisk analyse af jacket fundamenter til havvindmøller



Developing a Digital Twin for the Dynamic Analysis of Jacket Foundations for Offshore Wind Turbines

Udvikling af en digital tvilling for dynamisk analyse af jacket fundamenter til havvindmøller

Bachelor Thesis

June, 2025

Authors: Iven Henrik Meyer - s223826

Mia Steen Duus - s223832

Project period 03.02.2025-15.06.2025

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Cover photo: Vibeke Hempler, 2012

Published by: DTU, Department of Civil and Mechanical Engineering, Koppels Allé, Building 404, 2800 Kgs. Lyngby Denmark
www.construct.dtu.dk

Approval

This thesis has been prepared over four months at the Department of Civil and Mechanical Engineering, at the Technical University of Denmark, DTU, in partial fulfillment for the degree Bachelor of Science in Engineering, BSc Eng. The project has been developed with the assistance of AI technology, as described by Appendix E.

Iven Henrik Meyer - s223826



.....
Signature

15.06.2025

.....
Date

Mia Steen Duus - s223832



.....
Signature

15.06.2025

.....
Date

Acknowledgements

We would like to express our appreciation for our supervisor **Jan Becker Høgsberg**, Associate Professor at the Department of Civil and Mechanical Engineering, as well as our co-supervisor **Mikkel Tandrup Steffensen**, Postdoc at the Department of Civil and Mechanical Engineering for their guidance and support during the project in the completion of this report.

We would also like to extend our gratitude to Rambøll for the collaboration. Particularly, we thank **Dawid Augustyn** and **Vladimir Leble** for their time and expertise.

Abstract

Wind turbines experience on average two failures per year, each resulting in approximately 150 hours of downtime (Santelo et al., 2022). Hybrid Wind, a European research initiative, aims to investigate the conditions leading to failures in offshore wind turbines, particularly cold climate effects. To enhance damage detection and health monitoring, the project is advancing the development of digital twins (DTU, 2025). This thesis aims to contribute to the project by developing a simple digital twin for mode shape and natural frequency prediction for the jacket foundation of an offshore wind turbine.

The project aims to develop and validate simple finite element programs for dynamic analysis based on bar and beam elements. It investigates whether simpler models can accurately estimate the natural frequencies of a 3D Frame program. To facilitate comparison, four programs are developed: 2D Truss, 2D Frame, 3D Truss, and 3D Frame. From the finite element formulations, system stiffness and mass matrices for bar and beam elements are derived, forming the foundation for solving natural frequencies. The program implementation employs a parametric approach to construct input data for the jacket structure, enhancing usability.

Each program is applied in a natural frequency and modal analysis of a combined jacket-transition piece structure for an open-source 20 MW reference wind turbine. The analysis compares computation time and the approximation of natural frequencies against the 3D Frame program. Among the tested models, the 3D Truss program exhibits the most accurate approximation of natural frequencies when evaluating the first global mode for axial, bending, and torsional modes, with a maximum relative difference of 0.35%. Moreover, the 3D Truss model only requires 25% of the 3D Frame model's computation time.

The 2D Frame program provides a closer match to the 3D Frame program in bending modes 2 through 4, with relative differences of up to 1.8%. However, for the first global bending and axial mode has a relative difference of 16% and 13%, respectively. While the first modes deviate significantly, the 2D Frame model operates at just 10% of the computation time required by the 3D Frame program. By comparison, the 2D Truss program exhibits general deviations across different modes and is determined to be insufficient for accurately approximating natural frequencies.

Furthermore, the scope is expanded to include a simplified tower and RNA structure, which is implemented using beam elements across all programs. This implementation significantly enhances the approximation of natural frequencies determined by the 3D Frame program across all three models. Among them, the 3D Truss program provides the most reliable approximation, with a maximum relative difference of 1.5%. Additionally, the 3D Truss reduces computation time to 30–35% of the 3D Frame program.

This project demonstrates that a simplified 3D Truss model can effectively approximate natural frequencies for a combined wind turbine and support structure when compared to a 3D Frame model. The study highlights how the simplification of finite element models and use of bar elements can effectively reduce computation times, while maintaining reliability. Furthermore, the findings support the possibility of using simpler modeling approaches for preliminary design and experimental testing, offering practical alternatives to commercial finite element programs.

Symbols

$\mathbf{A}$	Transformation matrix between local and global coordinates	
A	Cross-sectional area	$[\text{m}^2]$
$\mathbf{B}$	Strain interpolation matrix	
B_i^d	Strain interpolation function at node i , direction d	
$\mathbf{D}$	Material property matrix	
E	Young's modulus	$[\text{Pa}]$
$\mathbf{F}_{ele}$	System element force vector	
$\mathbf{F}_{sys}$	System force vector	
G	Shear modulus	$[\text{Pa}]$
I	Second moment of area	$[\text{m}^4]$
J	Polar moment of area	$[\text{m}^4]$
$\mathbf{K}_{ele}$	System element stiffness matrix	
$\mathbf{K}_{sys}$	System stiffness matrix	
L	Element length	$[\text{m}]$
$\mathbf{M}_{ele}$	System element mass matrix	
$\mathbf{M}_{sys}$	System mass matrix	
M	Bending moment	$[\text{Nm}]$
$\mathbf{N}$	Displacement interpolation matrix	
N	Normal force	$[\text{N}]$
$\mathcal{N}_i^d$	Shape function for node i and direction d	
Q	Shear force	$[\text{N}]$
$\mathbf{d}$	Global element displacement vector	
$\mathbf{d}_e$	Local element displacement vector	
$\mathbf{d}_{ele}$	System element displacement vector	
$\mathbf{d}_{sys}$	System displacement vector	
$\mathbf{f}$	Global element force vector	
$\mathbf{f}_e$	Local element force vector	
$\mathbf{i}_{ele}$	Index array for global DOFs associated with a specific element	
i	Start node index of a single element in a coupled system	
j	End node index of a single element in a coupled system	

$\mathbf{k}$	Global element stiffness matrix	
$\mathbf{k}_e$	Local element stiffness matrix	
$\mathbf{m}$	Global element mass matrix	
$\mathbf{m}_e$	Local element mass matrix	
m_1, m_2	Local nodal moment at nodes 1 and 2	[Nm]
m_i, m_j	Global nodal moment at nodes i and j	[Nm]
n_{dof}	Total number of degrees of freedom in the global system	
n_{ele}	Total number of elements in the global system	
$\mathbf{p}_e$	Axial nodal force vector in the local coordinate system	
p	Axial distributed load	[N/m]
p_1, p_2	Axial nodal force at nodes 1 and 2	[N]
p_i, p_j	Axial nodal force in global x direction at nodes i and j	[N]
$\mathbf{q}_e$	Transverse nodal force vector in the local y direction	
q	Distributed transverse load in local y direction	[N/m]
q_1, q_2	Transverse nodal force in local y direction at nodes 1 and 2	[N]
q_i, q_j	Transverse nodal force in global y direction at nodes i and j	[N]
$\mathbf{r}_e$	Transverse nodal force vector in the local z direction	
r	Distributed transverse load in local z direction	[N/m]
r_1, r_2	Transverse nodal force in local z direction at nodes 1 and 2	[N]
r_i, r_j	Transverse nodal force in global z direction at nodes i and j	[N]
$\mathbf{s}_e$	Torsional nodal moment vector in the local coordinate system	
s_1, s_2	Torsional nodal moment in local axis at nodes 1 and 2	[Nm]
s_i, s_j	Torsional nodal moment in global x direction at nodes i and j	[Nm]
t	Time variable	[s]
$\mathbf{u}_e$	Axial nodal displacement vector in the local coordinate system	
u	Displacement field in local x direction	[m]
u_1, u_2	Local axial nodal displacement at nodes 1 and 2	[m]
u_i, u_j	Axial nodal displacement in global x direction at nodes i and j	[m]
$\mathbf{v}_e$	Transverse nodal displacement vector in the local y direction	[m]
v	Displacement field in local y direction	[m]
v_1, v_2	Transverse nodal displacement along the local y axis at nodes 1 and 2	[m]
v_i, v_j	Nodal displacement in global y direction at nodes i and j	[m]
$\mathbf{w}_e$	Transverse nodal displacement vector in the local z direction	
w	Displacement field in local z direction	[m]

w_1, w_2	Transverse nodal displacement along the local z axis at nodes 1 and 2	[m]
w_i, w_j	Nodal displacement in global z direction at nodes i and j	[m]
$\mathbf{x}_\ell$	Local x unit vector	
x_ℓ	Local coordinate along the beam axis	[m]
$\mathbf{y}_\ell$	Local y unit vector	
y_ℓ	Local coordinate perpendicular to the beam axis	[m]
$\mathbf{z}_\ell$	Local z unit vector	
z_ℓ	Local coordinate perpendicular to the beam axis	[m]
α	Inclination angle	[rad]
δ	Virtual displacement function	
$\boldsymbol{\varepsilon}$	Strain-state vector	
ε_x	Axial strain in local x direction	
γ	Torsional constant	[m ⁴]
κ	Section curvature	[1/m]
ν	Poisson's ratio	
$\boldsymbol{\phi}$	Torsional rotation vector	[rad]
ϕ_1, ϕ_2	Local torsional rotation at nodes 1 and 2	[rad]
ϕ_i, ϕ_j	Global torsional rotation at nodes i and j	[rad]
ϕ'	Derivative of torsional rotation	[rad/m]
ρ	Material density	[kg/m ³]
$\boldsymbol{\sigma}$	Stress-state vector	
θ_1, θ_2	Local nodal rotation at nodes 1 and 2	[rad]
θ_i, θ_j	Global nodal rotation at nodes i and j	[rad]
ω	Natural angular frequency	[rad/s]

List of Figures

1.1	Definition of offshore wind turbine and jacket.	2
2.1	Definition of the 2D bar element.	3
2.2	The bar elements' shape functions.	4
2.3	The 2D bar element transformation.	6
2.4	Definition of the 3D bar element.	9
2.5	3D vector transformation.	10
3.1	Definition of the 2D beam element.	12
3.2	The beam elements' shape functions.	13
3.3	Definition of the 3D beam element (torsion).	18
3.4	The 3D beam element with 12 DOFs.	22
4.1	Flow chart over program structure.	25
4.2	Process of building the beam model stiffness matrix.	27
5.1	Free-free bar element.	30
5.2	Relative frequency error against number of elements, 2D bar element.	33
5.3	Relative frequency error against number of elements, 3D bar element.	35
5.4	Clamped-free beam element.	36
5.5	Relative frequency error against number of elements, 2D beam element.	38
5.6	Relative frequency error against number of elements, 3D beam element.	41
6.1	Main geometric properties of the 20MW reference jacket.	42
6.2	3D Jacket and transition piece.	44
6.3	1st bending mode - 2D models.	45
6.4	1st bending mode - 3D models.	45
6.5	1st torsional mode.	46
6.6	1st and 2nd global bending mode - combined jacket, tower & RNA - 3D Frame.	48
6.7	3rd and 4th global bending mode - combined jacket, tower & RNA - 3D Frame.	48
6.8	Convergence of relative difference to value at 32 elements against number of subdivisions of elements.	51
6.9	Computation time against number of element subdivisions.	52
7.1	First eight eigenmodes of the jacket structure, ROSAP.	55

List of Tables

4.1	User-controlled data parameters.	28
4.2	User options for support types.	28
5.1	Material properties - free-free bar.	31
5.2	Relative errors of 2D longitudinal vibration for a free-free bar.	33
5.3	Nodal coordinates for 2D transformation test.	33
5.4	Validation of the 2D bar transformation matrix.	34
5.5	Relative errors of 3D longitudinal vibration for a free-free bar.	35
5.6	Nodal coordinates for 3D transformation test.	35
5.7	Validation of the 3D bar transformation matrix.	36
5.8	Material properties - clamped-free beam.	36
5.9	Relative errors of 2D transverse vibration for a clamped-free beam.	38
5.10	Validation of the 2D beam transformation matrix.	39
5.11	Material properties - clamped-free beam (torsion).	39
5.12	Relative errors of 3D torsional vibration for a clamped-free beam.	40
5.13	Validation of the 3D beam transformation matrix.	40
6.1	Geometric properties for leg and brace elements.	43
6.2	Material parameters for steel.	43
6.3	Transition piece properties.	43
6.4	Mass of models.	43
6.5	Global bending mode frequencies [Hz] for jacket with corresponding system mode numbers.	44
6.6	Global axial mode frequencies [Hz] for jacket with corresponding system mode numbers.	46
6.7	Global torsional mode frequencies [Hz] for jacket with corresponding system mode numbers.	46
6.8	Tower geometry.	47
6.9	Tower and RNA properties.	47
6.10	Global tower mode frequencies [Hz].	47
6.11	Global bending mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.	48
6.12	Global axial mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.	49
6.13	Global torsional mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.	49
6.16	Convergence results for 2D Frame.	50
6.17	Convergence results for 3D Frame.	51
7.1	Natural frequency for first 10 modes, ROSAP.	54
7.2	Comparison of frequencies to global bending modes in ROSAP and 3D Frame, with corresponding mode number.	55

Contents

1	Introduction	1
2	The Bar Element	3
2.1	The 2D Bar Element	3
2.2	Finite Element Formulation for Axial Vibration	5
2.3	Assembling System Matrices for the 2D Bar	6
2.4	The 3D Bar Element	9
2.5	Assembling System Matrices for the 3D Bar	9
3	The Beam Element	12
3.1	The 2D Beam Element	12
3.2	Finite Element Formulation for Bending Vibration	14
3.3	Assembling System Matrices for the 2D Beam	16
3.4	2D Beam Element with Axial and Transverse Effects	16
3.5	The 3D Beam Element	18
3.6	Finite Element Formulation for Torsional Vibration	19
3.7	Assembling System Matrices for the 3D Beam	19
3.8	3D Beam Element with Axial, Transverse and Torsional Effects	20
3.9	Mode Shapes and Natural Frequencies	23
4	Development of the FE program	24
4.1	Objective and Application	24
4.2	Program Structure	24
4.3	Expansion of the Project Scope	29
5	Program Validation and Testing	30
5.1	2D Truss	30
5.2	3D Truss	34
5.3	2D Frame	36
5.4	3D Frame	39
6	Computational Analysis of Jacket Models	42
6.1	Natural Frequencies of the Jacket Structure	42
6.2	Natural Frequencies of the Jacket-Wind Turbine System	46
6.3	Computation Time	49
6.4	Convergence of the Frame Models	50
7	Evaluation of the Computational Models	53
7.1	Program Assessment for Jacket Frequency Analysis	53
7.2	Program Assessment for Jacket-Turbine Frequency Analysis	54
7.3	Model Benchmarking and Computation Time	56
7.4	Convergence of the Frame Models	57
8	Conclusion	58
9	Future Work	59

Bibliography	61
A Lagrange Multipliers	63
A.1 Coupling of Jacket Foundation and Tower Model in 2D	63
A.2 Coupling of Jacket Foundation and Tower Model in 3D	64
B Jacket Design	67
C List of Mode Shapes	70
D Digital Appendix	72
D.1 Programs	72
D.2 Mode Shape Plots	72
D.3 Validation Files	73
E Use of AI Technology	74

1 Introduction

In 2022, EU countries adopted the 'Trans-European Networks for Energy' policy, a cross-border framework dedicated to offshore grid planning. In continuation on this, in 2024, EU nations set an ambitious goal to achieve an offshore wind capacity of 356-366 GW by 2050 (European Commission, 2025). However, wind turbines currently experience an average of two failures per year, each resulting in approximately 150 hours of downtime. (Santelo et al., 2022). This highlights a critical need for research into the conditions leading to these failures. Furthermore, the development of robust health and performance monitoring systems is essential for early damage detection, ensuring the reliability and efficiency of offshore wind energy.

The Hybrid Wind project

Hybrid Wind is a major European research project, which aims to investigate the effect of cold climate on turbine conditions. One of the main objectives of the project is to advance damage detection and increase the robustness to weather effects. The objectives are to be expected to be achieved through the development of digital twins for both simulations and real time monitoring, as well as experimental work to test physical components under systematic variation of conditions. The need for research of temperature conditions is due to the unknown impact of temperature on critical loads and thereby risk of exceeding design parameters (DTU, 2025).

Project Main Objective

This project aims to contribute to the Hybrid Wind project by creating an 'in-house' finite element (FE) program to determine natural frequencies of a jacket foundation. Serving as a simplified digital twin, the program will simulate natural frequencies and mode shapes to support experimental testing and preliminary design. The objective for the developed program is to be a user-friendly, cost-effective tool. The program will utilize a parametric approach to generate input data to the jacket structure, minimizing the required for application.

Various FE software solutions for dynamic analysis are available, predominantly commercial platforms that require expertise in their operation and significant computational time for practical application. The developed program in this project is designed to substantially reduce the time required for application and computation, while efficiently approximating the natural frequencies of the jacket structure. The program will be developed in Python, an open-source programming language known for its simple syntax, making it accessible to users with only limited prior coding experience.

The project will investigate whether simpler models can be used as alternatives to a full-scale 3D Frame model, while providing reliable results. The project scope will include developing programs for simulation in 2D and 3D utilizing finite element formulations for bar and beam elements, respectively. This will result in a comparative analysis of the four individual programs (2D Truss, 2D Frame, 3D Truss & 3D Frame). A key limitation of the project is the absence of laboratory and operational data for validating the results. The analysis will include a comparison to a dynamic analysis performed by Pontow et al. (2017) in Ramboll's Offshore Structural Analysis Program (ROSAP), a tool employed for structural design and monitoring within Ramboll (Ramboll, 2025).

The Jacket Foundation

The jacket foundation is one type of support structure for offshore wind turbines, which is a permanently fixed lattice structure, anchored to the seabed using piles. Jacket structures typically

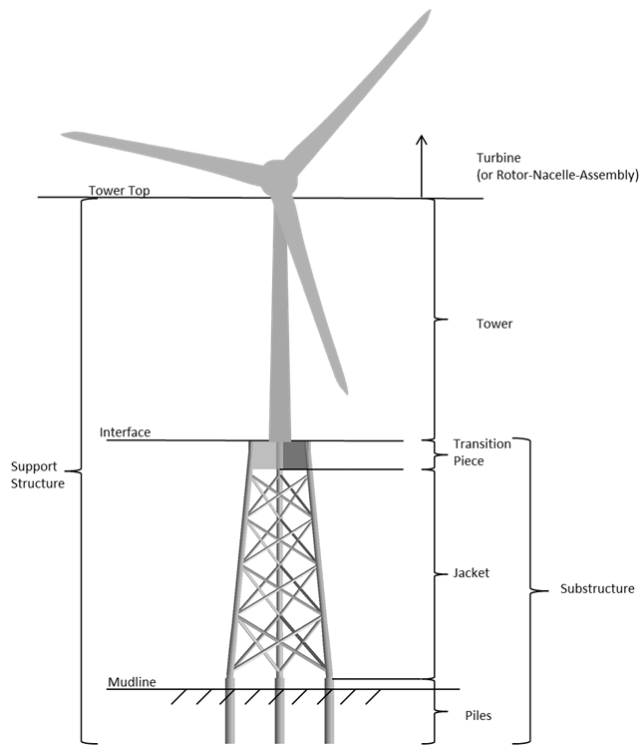


Figure 1.1: Definition of offshore wind turbine and jacket. Pontow, S., Kaufer, D., Shirzadeh, R. & Kühn M. (2017), *Design Solution for a Support Structure Concept for future 20MW*, IN-NWIND.EU, p. 8.

have three or four sides, each with a corresponding number of fixtures and legs. The leg elements are reinforced by X-braces, forming the lattice design. This project specifically focuses on the four-sided jacket structure.

Figure 1.1 shows the design of the collected support structure for a wind turbine, where the jacket is built of 4 levels of X-braces. The main geometric parameters of the jacket structure include the width at the mudline, the width at the top, and the overall height of the jacket. For this project, the initial analysis will include the combined system of the jacket and transition piece. Additionally, the tower and Rotor-Nacelle-Assembly will be added to enable a comparative analysis of the natural frequencies between the complete system and the combined jacket-transition piece structure.

Thesis structure

The report is structured into four main parts, following the progression of the project. The first part introduces the finite element theory for bar (Chapter 2) and beam elements (Chapter 3). This theoretical framework provides the foundational principles for the development of the truss and frame programs, respectively.

The second part details the implementation and program structure (Chapter 4). This part will also include a validation process comparing the computation of mass and stiffness matrices with the theoretical solutions outlined in the first part of the report (Chapter 5).

In the third part, an assessment of the developed programs will be performed. Firstly, through a natural frequency and modal analysis of a jacket structure (Chapter 6). Secondly, through the comparative analysis and evaluation of the results from the analysis (Chapter 7). This will include a comparison to the natural frequency analysis presented in the report by Pontow et al. (2017).

Finally, the conclusions drawn from the assessment of the programs are presented, including an evaluation of the defined objectives (Chapter 8). Additionally, a brief perspective on future work and potential opportunities for expanding the project is discussed (Chapter 9).

2 The Bar Element

The foundation of any finite element program lies in the definition and formulation of its elements. In this chapter, the theoretical background of the bar element is presented, using the principle of virtual work. The resulting element stiffness and mass matrix are formulated in both 2D and the 3D space. Ultimately, the assembly of these matrices into global system matrices is demonstrated. This process forms the basis for deriving natural frequencies through an eigenvalue analysis.

2.1 The 2D Bar Element

The simplest element in finite element analysis (FEA) is the bar element illustrated in Figure 2.1, which possesses one degree of freedom (DOF) per node. The bar element is a one-dimensional element that only carries axial loads. The theory outlined in Logan (2021) serves as the foundation for the theory of the bar element.

The element is characterized by two nodes, namely a start node (1) and an end node (2). It has a defined length L , cross-sectional area A , density ρ , and Young's modulus E . The bar element is shown schematically in Figure 2.1, with nodal displacements u_1 and u_2 and axial nodal forces p_1 and p_2 along the local x_ℓ . The local x -axis is axial to the bar element, pointing from the start node (1) to the end node (2). The nodal displacement vector $\mathbf{u}_e$ and the nodal force vector $\mathbf{p}_e$ for the element in the local coordinate system are given by

$$\mathbf{u}_e = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_e \quad \mathbf{p}_e = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}_e \quad (2.1)$$

It should be noted that the element is not capable of withstanding shear and bending forces, and any effect of transverse displacement is disregarded.

Shape Functions

The local axial displacements $u(x_\ell)$ of the element are shown to vary linearly. Thus, the displacements can be interpolated using the shape functions $\mathcal{N}_1^u(x_\ell)$ and $\mathcal{N}_2^u(x_\ell)$ and the nodal displacements u_1 and u_2 . This results in the linear relationship

$$u(x_\ell) = \mathcal{N}_1^u(x_\ell) u_1 + \mathcal{N}_2^u(x_\ell) u_2 \quad (2.2)$$

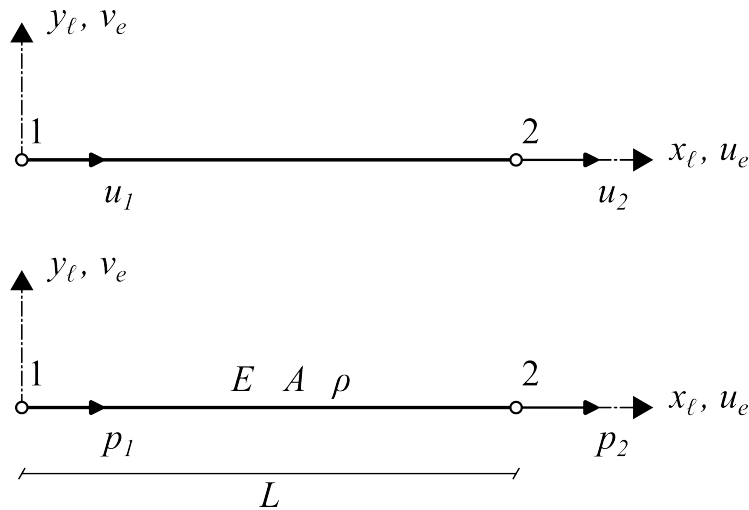


Figure 2.1: Definition of the 2D bar element.

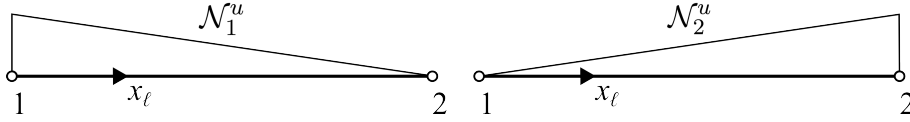


Figure 2.2: The bar elements' shape functions.

with the shape functions

$$\mathcal{N}_1^u(x_l) = 1 - \frac{x_l}{L} \quad \text{and} \quad \mathcal{N}_2^u(x_l) = \frac{x_l}{L} \quad (2.3)$$

It is essential that the shape functions in Figure 2.2 meet the criteria of assuming the value 1 at one end and value 0 at the opposing end. Eq. (2.2) describes the axial displacement field of the element, which can be written in a matrix format as

$$u = [\mathcal{N}_1^u \quad \mathcal{N}_2^u] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_e \quad (2.4)$$

which can be formulated in compact form as

$$u = \mathbf{N}_u \mathbf{u}_e \quad (2.5)$$

where $\mathbf{N}_u$ denotes the displacement interpolation matrix for the axial displacement.

Stress and Strain

Axial strain is defined as the relative deformation to a reference position by

$$\varepsilon_x = \frac{du}{dx_l} \quad (2.6)$$

which is expressed via the spatial derivative. In general, the strain can contain several dimensions. Therefore, the strain state is introduced in a matrix format. By substituting Eq. (2.5) into Eq. (2.6), the strain vector becomes

$$\boldsymbol{\varepsilon} = [\varepsilon_x] = \left[\frac{d}{dx} \right] [\mathcal{N}_1^u \quad \mathcal{N}_2^u] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_e \quad (2.7)$$

The strain interpolation matrix, denoted by $\mathbf{B}$, is introduced. It is important to note that the defined interpolation matrix $\mathbf{B}_u$ for the 2D bar element is constant and independent of x , and can be expressed by

$$\mathbf{B}_u = [B_1^u \quad B_2^u] = \left[\frac{d}{dx} \right] [\mathcal{N}_1^u \quad \mathcal{N}_2^u] \quad (2.8)$$

whereby Eq. (2.7) can be formulated in compact form as

$$\boldsymbol{\varepsilon} = \mathbf{B}_u \mathbf{u}_e \quad (2.9)$$

The section force, denoted by N , is defined as the normal force and is the only section force acting in the 2D bar element. The stress state $\boldsymbol{\sigma}$ can contain several stress quantities and is therefore introduced as a vector. The stress is described by the forces present during deformation as

$$\boldsymbol{\sigma} = [N] = [EA] [\varepsilon_x] \quad (2.10)$$

where

$$\mathbf{D} = [EA] \quad (2.11)$$

is the material-stiffness matrix and EA is an expression for the axial elastic stiffness. Hence, the stress state of the element is expressed in compact form as

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon} = \mathbf{D} \mathbf{B}_u \mathbf{u}_e \quad (2.12)$$

employing the strain interpolation matrix from Eq. (2.7).

2.2 Finite Element Formulation for Axial Vibration

The axial motion of an elastic bar is described by the differential equation (Krenk, 2018)

$$(EAu')' - \rho A \ddot{u} + p = 0 \quad (2.13)$$

where the axial displacement $u(x, t)$ and axial distributed load $p(x, t)$ are dependent on both time, t and the local coordinate x along the length of the bar element, running from $0 \rightarrow L$ as shown in Figure 2.1. Additionally, ρA is the mass per unit length. The notation is defined by

$$(\cdot)' = \frac{\partial}{\partial x} \quad (\dot{\cdot}) = \frac{\partial}{\partial t} \quad (2.14)$$

where primes denote differentiation with respect to x and dots denote differentiation with respect to time.

2.2.1 The Principle of Virtual Work

The principle of virtual work is employed. Hence, Eq. (2.13) is multiplied by a virtual displacement field δu and subsequently integrated over the length of the bar

$$\int_0^L \delta u \{ (EAu')' - \rho A \ddot{u} + p \} dx = 0 \quad (2.15)$$

The first term in Eq. (2.15) is integrated once, using integration by parts. Thereby, establishing symmetry in the first term, by the following integration

$$\int_0^L \delta u (EAu')' dx = [\delta u (EA u')]_0^L - \int_0^L \delta u' EA u' dx \quad (2.16)$$

Eq. (2.16) is substituted into Eq. (2.15), and terms in the weak form equation are separated into

$$\int_0^L \delta u' EA u' dx + \int_0^L \delta u \rho A \ddot{u} dx = [\delta u (EA u')]_0^L + \int_0^L \delta u p dx \quad (2.17)$$

where the equivalent nodal loads and isolated nodal loads are collected on the right hand side.

2.2.2 Implementing Shape Functions

The virtual displacement and virtual strain is described by replacing the nodal displacements with virtual nodal displacements δu in Eq. (2.4) and Eq. (2.9) leading to

$$\delta u(x, t) = \mathbf{N}_u \delta \mathbf{u}_e = (\delta \mathbf{u}_e)^T \mathbf{N}_u^T \quad (2.18)$$

$$\delta u'(x, t) = \mathbf{B}_u \delta \mathbf{u}_e = (\delta \mathbf{u}_e)^T \mathbf{B}_u^T \quad (2.19)$$

The time derivative of the displacement field in Eq. (2.4) yields

$$\ddot{u}(x, t) = \mathbf{N}_u \ddot{\mathbf{u}}_e = [\mathcal{N}_1^u \quad \mathcal{N}_2^u] \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \end{bmatrix}_e \quad (2.20)$$

The axial displacement u is expressed in terms of Eq. (2.4) and the strain for the bar element is described using Eq. (2.9). Additionally Eq. (2.18)-(2.20) are substituted into Eq. (2.17). The work equation is reformulated as

$$\begin{aligned} & \int_0^L (\delta \mathbf{u}_e)^T \mathbf{B}_u^T EA \mathbf{B}_u \mathbf{u}_e dx + \int_0^L (\delta \mathbf{u}_e)^T \mathbf{N}_u^T \rho A \mathbf{N}_u \ddot{\mathbf{u}}_e dx \\ &= [(\delta \mathbf{u}_e)^T \mathbf{N}_u^T N]_0^L + \int_0^L (\delta \mathbf{u}_e)^T \mathbf{N}_u^T p dx \end{aligned} \quad (2.21)$$

where EAu' defines the normal force N in the square brackets of Eq. (2.17).

2.2.3 Deriving the Equation of Motion

All terms are collected on the left-hand side in order to factorize the equation

$$(\delta \mathbf{u}_e)^T \left\{ \left(\int_0^L \mathbf{B}_u^T E A \mathbf{B}_u dx \right) \mathbf{u}_e + \left(\int_0^L \mathbf{N}_u^T \rho A \mathbf{N}_u dx \right) \ddot{\mathbf{u}}_e - [\mathbf{N}_u^T N]_0^L - \int_0^L \mathbf{N}_u^T p dx \right\} = 0 \quad (2.22)$$

with the virtual displacement $(\delta \mathbf{u}_e)^T$ as a common factor. Since the virtual displacement $\delta \mathbf{u}_e \neq 0$, Eq. (2.22) can be reformulated using the zero-product property by

$$\left(\int_0^L \mathbf{B}_u^T E A \mathbf{B}_u dx \right) \mathbf{u}_e + \left(\int_0^L \mathbf{N}_u^T \rho A \mathbf{N}_u dx \right) \ddot{\mathbf{u}}_e - [\mathbf{N}_u^T N]_0^L - \int_0^L \mathbf{N}_u^T p dx = 0 \quad (2.23)$$

The external nodal forces are moved back to the right-hand side. This reveals the expression

$$\left(\int_0^L \mathbf{B}_u^T E A \mathbf{B}_u dx \right) \mathbf{u}_e + \left(\int_0^L \mathbf{N}_u^T \rho A \mathbf{N}_u dx \right) \ddot{\mathbf{u}}_e = [\mathbf{N}_u^T N]_0^L + \int_0^L \mathbf{N}_u^T p dx \quad (2.24)$$

for the dynamic response under a time-dependent external force. By identifying the coefficients for both the displacement and acceleration, the element stiffness matrix $\mathbf{k}_e$ and the element mass matrix $\mathbf{m}_e$ are defined by the integrals

$$\mathbf{k}_e = \int_0^L \mathbf{B}_u^T E A \mathbf{B}_u dx = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.25)$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}_u^T \rho A \mathbf{N}_u dx = \frac{\rho AL}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (2.26)$$

which in this case are evaluated analytically. Furthermore, introducing the external force vector $\mathbf{p}_e$, as defined in Eq. (2.1), allows the dynamic response equation Eq. (2.24) to be expressed as

$$\mathbf{k}_e \mathbf{u}_e + \mathbf{m}_e \ddot{\mathbf{u}}_e = \mathbf{p}_e \quad (2.27)$$

finalizing the derivation of the finite element formulation for axial motion.

2.3 Assembling System Matrices for the 2D Bar

2.3.1 Element Transformation

The theory introduced so far is only valid in local coordinates. Subsequently, when dealing with more complex structures, the model must be able to accommodate arbitrarily positioned elements in a two-dimensional space utilizing global coordinates. The transformation of the bar element is formulated based on Nielsen et al. (2019) and visualized in Figure 2.3.

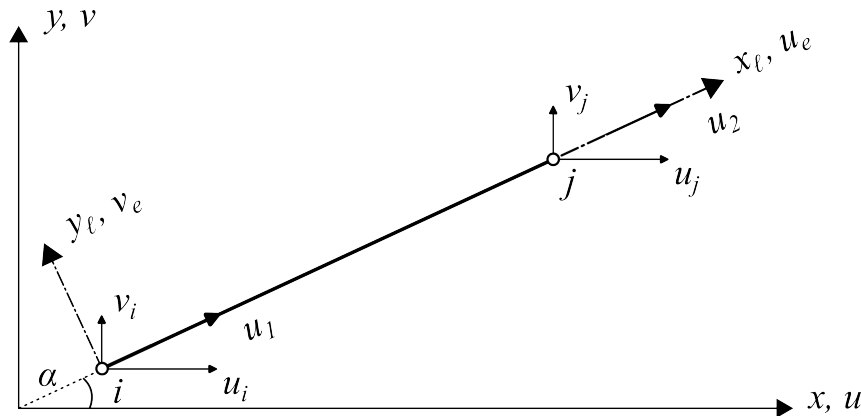


Figure 2.3: The 2D bar element transformation.

Using standard trigonometric relations between the local and global coordinate systems, the displacement u_e in the element's local axis can be expressed in terms of the global displacements u and v as

$$u_e = u \cos \alpha + v \sin \alpha \quad (2.28)$$

where α is the angle between the element axis and the global x -axis, as shown in Figure 2.3. Furthermore, it can be beneficial to introduce

$$\begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \frac{1}{L} \quad (2.29)$$

where n_x and n_y are the components of the unit direction vector. The global coordinates of nodes i and j are denoted by x_i, y_i and x_j, y_j . In order to utilize the mass matrix in global coordinates, it is necessary to expand the mass matrix from Eq. (2.26) into a 4x4 matrix. This can be achieved by expanding the interpolation matrix $\mathbf{N}_u$ to a 2x4 matrix, resulting in

$$\mathbf{N}_u = \begin{bmatrix} \mathcal{N}_1^u & 0 & \mathcal{N}_2^u & 0 \\ 0 & \mathcal{N}_1^u & 0 & \mathcal{N}_2^u \end{bmatrix} \quad (2.30)$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}_u^T \rho A \mathbf{N}_u dx = \frac{\rho AL}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \quad (2.31)$$

$$\mathbf{k}_e = \int_0^L \mathbf{B}_u^T EA \mathbf{B}_u dx = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.32)$$

The relationship between local and global nodal displacements can be expressed by

$$\begin{bmatrix} u_1 \\ 0 \\ u_2 \\ 0 \end{bmatrix}_e = \begin{bmatrix} n_x & n_y & 0 & 0 \\ -n_y & n_x & 0 & 0 \\ 0 & 0 & n_x & n_y \\ 0 & 0 & -n_y & n_x \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{bmatrix} \Rightarrow \mathbf{u}_e = \mathbf{A} \mathbf{d} \quad (2.33)$$

introducing a transformation matrix $\mathbf{A}$. The transformation matrix is applied to the load vector accordingly

$$\begin{bmatrix} p_1 \\ 0 \\ p_2 \\ 0 \end{bmatrix}_e = \begin{bmatrix} n_x & n_y & 0 & 0 \\ -n_y & n_x & 0 & 0 \\ 0 & 0 & n_x & n_y \\ 0 & 0 & -n_y & n_x \end{bmatrix} \begin{bmatrix} p_i \\ q_i \\ p_j \\ q_j \end{bmatrix} \Rightarrow \mathbf{p}_e = \mathbf{A} \mathbf{f} \quad (2.34)$$

In Eq. (2.33) and Eq. (2.34), the vectors $\mathbf{d}$ and $\mathbf{f}$ are introduced to represent the nodal displacements and nodal forces, in the global coordinate system along the global x - and y -axes. Subsequently, Eq. (2.27) is recalled and multiplied by a virtual displacement

$$\delta \mathbf{u}_e^T (\mathbf{k}_e \mathbf{u}_e + \mathbf{m}_e \ddot{\mathbf{u}}_e - \mathbf{p}_e) = 0 \quad (2.35)$$

Utilizing the transformation to global coordinates, Eq. (2.33) and Eq. (2.34) are substituted into Eq. (2.35)

$$\delta \mathbf{d}^T (\mathbf{A}^T \mathbf{k}_e \mathbf{A} \mathbf{d} + \mathbf{A}^T \mathbf{m}_e \mathbf{A} \ddot{\mathbf{d}} - \mathbf{A}^T \mathbf{p}_e) = 0 \quad (2.36)$$

Given that the virtual nodal displacement, $\delta \mathbf{d}^T \neq 0$, the equilibrium condition must hold. This leads to the final form

$$\mathbf{A}^T \mathbf{k}_e \mathbf{A} \mathbf{d} + \mathbf{A}^T \mathbf{m}_e \mathbf{A} \ddot{\mathbf{d}} = \mathbf{A}^T \mathbf{p}_e \quad (2.37)$$

of the equation of motion for a single element in global coordinates. Furthermore, the following relationships obtained

$$\mathbf{k} = \mathbf{A}^T \mathbf{k}_e \mathbf{A}, \quad \mathbf{m} = \mathbf{A}^T \mathbf{m}_e \mathbf{A}, \quad \mathbf{d} = \mathbf{A}^T \mathbf{u}_e \quad \mathbf{f} = \mathbf{A}^T \mathbf{p}_e \quad (2.38)$$

for the stiffness and mass matrices, as well as the force vector.

2.3.2 Assembling the System Matrix

In order to formulate an equation for a system of elements, it is necessary to couple the element equation from Eq. (2.37). The system stiffness matrix for a single element, denoted by $\mathbf{K}_{ele}$, is hereby introduced. The system element stiffness matrix $\mathbf{K}_{ele}$, is a square matrix where the number of columns and rows corresponds to the total number of global DOFs (n_{dof}) in the system. The global element stiffness matrix $\mathbf{k}(i, j)$, for an element between node i and j , is placed at indices $\mathbf{i}_{ele}$ using

$$\mathbf{K}_{ele}(\mathbf{i}_{ele}(i, j), \mathbf{i}_{ele}(i, j)) = \mathbf{k}(i, j) \quad (2.39)$$

$$i = 1, 2, \dots, n_{dof}, \quad j = 1, 2, \dots, n_{dof} \quad (2.40)$$

where $\mathbf{i}_{ele}$ contains the global DOF numbers associated with the specific element

$$\mathbf{i}_{ele} = [2i - 1 \quad 2i \quad 2j - 1 \quad 2j]_{ele} \quad (2.41)$$

for an element with a total of four DOFs. The elements' start and end nodes are denoted by the global node numbers i and j . Each of the global element stiffness matrices are added together in order to obtain the global system matrix

$$\mathbf{K}_{sys} = \sum_{ele=1}^{n_{ele}} \mathbf{K}_{ele} \quad (2.42)$$

with n_{ele} being the total number of elements in the global system. Having established the system matrix, $\mathbf{K}_{sys}$, the system mass matrix $\mathbf{M}_{sys}$ is introduced using the same approach of combining element mass matrices

$$\mathbf{M}_{ele}(\mathbf{i}_{ele}(i, j), \mathbf{i}_{ele}(i, j)) = \mathbf{m}(i, j) \quad (2.43)$$

$$i = 1, 2, \dots, n_{dof} \quad j = 1, 2, \dots, n_{dof} \quad (2.44)$$

The system element mass matrices are then added to form the global system mass matrix

$$\mathbf{M}_{sys} = \sum_{ele=1}^{n_{ele}} \mathbf{M}_{ele} \quad (2.45)$$

The force, displacement and acceleration vectors of each element are combined to form the coupled element equations. Each element vector is expanded into a global element vector of length n_{dof}

$$\mathbf{F}_{ele}(\mathbf{i}_{ele}) = \mathbf{f}(i, j), \quad \mathbf{d}_{ele}(\mathbf{i}_{ele}) = \mathbf{d}(i, j), \quad \ddot{\mathbf{d}}_{ele}(\mathbf{i}_{ele}) = \ddot{\mathbf{d}}(i, j) \quad (2.46)$$

by allocating the force, displacement and acceleration to the corresponding DOFs respectively. The system vectors are obtained by summing the vectors of the system elements

$$\mathbf{F}_{sys} = \sum_{ele=1}^{n_{ele}} \mathbf{F}_{ele}, \quad \mathbf{d}_{sys} = \sum_{ele=1}^{n_{ele}} \mathbf{d}_{ele}, \quad \ddot{\mathbf{d}}_{sys} = \sum_{ele=1}^{n_{ele}} \ddot{\mathbf{d}}_{ele} \quad (2.47)$$

The dynamic response of a single element is described by the element matrices in global coordinates in Eq. (2.37), which extends to a system of elements by employing the global system vectors and matrices defined by Eq. (2.42), (2.45) and (2.47)

$$\mathbf{K}_{sys} \mathbf{d}_{sys} + \mathbf{M}_{sys} \ddot{\mathbf{d}}_{sys} = \mathbf{F}_{sys} \quad (2.48)$$

which describes the dynamic response of a global system.

2.4 The 3D Bar Element

The 3D bar element, shown in Figure 2.4, is derived from an expansion of the 2D bar element. The theoretical construct is identical to that previously described in Section 2.1 and 2.2. Hence, the element's transformation from 2D to 3D space results in the acquisition of an additional local axis z_ℓ (Logan, 2021).

The shape functions $\mathcal{N}_1^u(x_\ell)$ and $\mathcal{N}_2^u(x_\ell)$ are identical to those used in the 2D case defined by Eq. (2.3) and employed for defining the mass and stiffness matrices. The element mass and stiffness matrices are shown in Eq. (2.25) and Eq. (2.26). Even if an element is considered in 3D space, the incorporation of the z -coordinate has no effect on the displacement function u compared to that in 2D space.

2.5 Assembling System Matrices for the 3D Bar

2.5.1 Element Transformation

The 3D bar element possesses the capability to be arbitrarily placed in the 3D space. Therefore, similarly to the 2D bar element, a transformation matrix is required to transform the node coordinates from the local system to the global system. The coordinate transformation between the local to global system is given by

$$u_e = u \cos \alpha_x + v \cos \alpha_y + w \cos \alpha_z \quad (2.49)$$

As previously mentioned in Section 2.3.1, the mass matrix, and consequently the stiffness matrix, must be expanded to 6x6 matrices to align with 3D coordinates. This is achieved by expanding the interpolation matrix $\mathbf{N}_u$ to a 3x6 matrix as

$$\mathbf{N}_u = \begin{bmatrix} \mathcal{N}_1^u & 0 & 0 & \mathcal{N}_2^u & 0 & 0 \\ 0 & \mathcal{N}_1^u & 0 & 0 & \mathcal{N}_2^u & 0 \\ 0 & 0 & \mathcal{N}_1^u & 0 & 0 & \mathcal{N}_2^u \end{bmatrix} \quad (2.50)$$

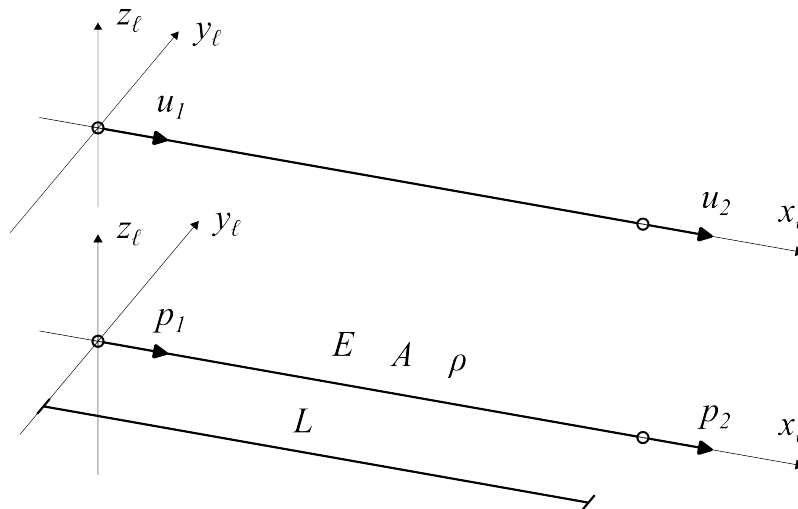


Figure 2.4: Definition of the 3D bar element.

which evidently also results in an expansion of both the displacement and the force vector by

$$\mathbf{u}_e = [u_1 \ 0 \ 0 \ u_2 \ 0 \ 0]^T, \quad \mathbf{p}_e = [p_1 \ 0 \ 0 \ p_2 \ 0 \ 0]^T \quad (2.51)$$

An element transformation matrix $\mathbf{A}$ is defined by the element's orientation in the 3D space, using the following geometric assumptions

- The local x -axis ($\mathbf{x}_\ell$) is defined as a unit vector from the start node to the end node of the element.
- The local y -axis is always perpendicular to the local x -axis ($\mathbf{x}_\ell$) and global z ($\mathbf{z} = [0, 0, 1]^T$), calculated via the cross product $\mathbf{y}_\ell = \mathbf{z} \times \mathbf{x}_\ell$ to ensure orthogonality.

In the special case where the element lies parallel to the global z -axis (i.e., $\mathbf{x}_\ell \parallel \mathbf{z}$), the cross product yields a zero vector. In this case, the local $\mathbf{y}_\ell$ is defined as the unit global y -axis: $\mathbf{y}_\ell = [0, 1, 0]^T$.

- The local z -axis, denoted by z_ℓ , is defined as a vector perpendicular to both the local x -axis ($\mathbf{x}_\ell$) and the local y -axis ($\mathbf{y}_\ell$), calculated using the cross product $\mathbf{z}_\ell = \mathbf{x}_\ell \times \mathbf{y}_\ell$

The transformation matrix $\mathbf{A}$ takes the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \quad \text{with} \quad \mathbf{T} = \begin{bmatrix} \mathbf{x}_\ell^T \\ \mathbf{y}_\ell^T \\ \mathbf{z}_\ell^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_\ell^x & \mathbf{x}_\ell^y & \mathbf{x}_\ell^z \\ \mathbf{y}_\ell^x & \mathbf{y}_\ell^y & \mathbf{y}_\ell^z \\ \mathbf{z}_\ell^x & \mathbf{z}_\ell^y & \mathbf{z}_\ell^z \end{bmatrix} \quad (2.52)$$

where each row of $\mathbf{T}$ corresponds to the direction cosines of the local axes expressed in the global coordinate system. The element transformation is visualized in Figure 2.5.

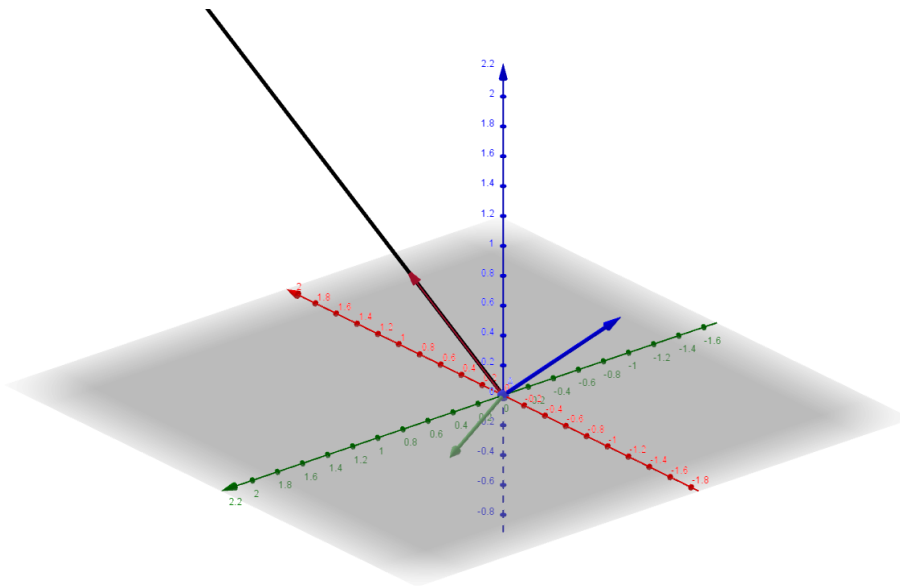


Figure 2.5: 3D vector transformation.

The relationship between local and global nodal displacements can be expressed by

$$\begin{bmatrix} u_1 \\ 0 \\ 0 \\ u_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{bmatrix} \quad (2.53)$$

using the transformation matrix $\mathbf{A}$ from Eq. (2.52). The same applies to the load vector $\mathbf{p}_e$. Recalling Eq. (2.37), the corresponding relationships between the global and local element quantities are given by Eq. (2.38) with the matrices and vectors defined in the global system.

2.5.2 Assembling the System Matrix

The global system matrices are assembled as outlined in Section 2.3.2, using index array

$$\mathbf{i}_{ele} = [3i - 2 \quad 3i - 1 \quad 3i \quad 3j - 2 \quad 3j - 1 \quad 3j]_{ele} \quad (2.54)$$

for an element with a total of six global DOFs.

3 The Beam Element

Following the formulation of the bar element and derivation of axial displacements, the beam element is introduced, accounting for bending behavior of the element. This provides the foundation for the subsequent chapters, in which the dynamic response due to transverse forces and torsional moments is examined. Ultimately, this results in the formulation of the eigenvalue problem and the determination of natural frequencies.

3.1 The 2D Beam Element

The 2D beam element in Figure 3.1 possesses two DOFs at each node and is subjected to transverse displacement and rotation. Hence the element has the capacity to transfer shear forces and bending moments. The considered beam element in Figure 3.1 introduces the second moment of area I . In the local coordinate system, the nodal displacement vector $\mathbf{v}_e$ and the nodal force vector $\mathbf{q}_e$ are expressed by

$$\mathbf{v}_e = [v_1 \quad \theta_1 \quad v_2 \quad \theta_2]^T_e \quad (3.1)$$

$$\mathbf{q}_e = [q_1 \quad m_1 \quad q_2 \quad m_2]^T_e \quad (3.2)$$

The transverse nodal displacements are denoted by v and the rotations by θ , while q and m represent the transverse forces and moments, respectively.

Shape Functions

The variation in transverse displacements are selected to be a function $v(x_\ell)$ that satisfies the Euler–Bernoulli equation. There are four DOFs in total and the function must satisfy the conditions of displacement and slope continuity (Logan, 2021). For the beam element a third order polynomial is chosen as

$$v(x_\ell) = a_1 x_\ell^3 + a_2 x_\ell^2 + a_3 x_\ell + a_4 \quad (3.3)$$

The following step involves expressing the nodal DOFs via the displacement function $v(x)$, achieved by substituting either $v(0)$ or $v(L)$ accordingly. For the polynomial in Eq. (3.3), the nodal displace-

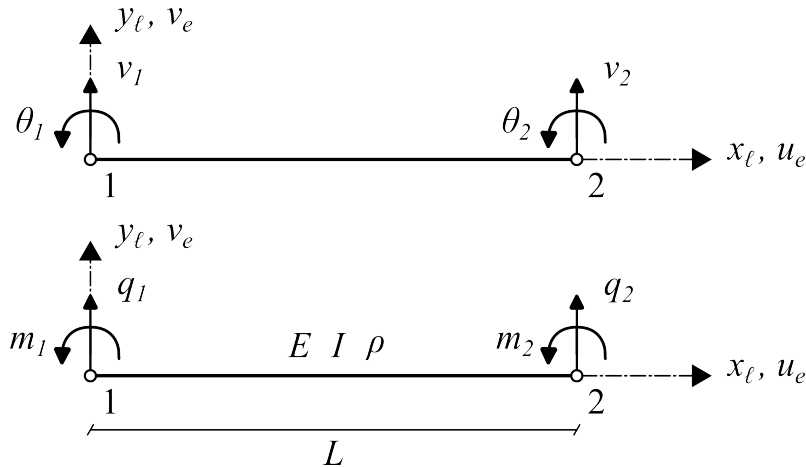


Figure 3.1: Definition of the 2D beam element.

ments and rotations are derived as

$$v(0) = v_1 = a_4 \quad (3.4)$$

$$\frac{dv(0)}{dx_\ell} = \theta_1 = a_3 \quad (3.5)$$

$$v(L) = v_2 = a_1 L^3 + a_2 L^2 + a_3 L + a_4 \quad (3.6)$$

$$\frac{dv(L)}{dx_\ell} = \theta_2 = a_1 L^2 + a_2 L + a_3 \quad (3.7)$$

Upon solving the system of equations (Eq. (3.4) - (3.7)), the following results for a_i are obtained

$$a_1 = \frac{2}{L^3}(v_1 - v_2) + \frac{1}{L^2}(\theta_1 + \theta_2), \quad (3.8)$$

$$a_2 = -\frac{3}{L^2}(v_1 - v_2) - \frac{1}{L}(2\theta_1 + \theta_2), \quad (3.9)$$

$$a_3 = \theta_1, \quad (3.10)$$

$$a_4 = v_1 \quad (3.11)$$

When substituting Eq. (3.8) - (3.11) into Eq. (3.3), the displacement function can be expressed in a matrix format as

$$v = \begin{bmatrix} \mathcal{N}_1^v & \mathcal{N}_1^\theta & \mathcal{N}_2^v & \mathcal{N}_2^\theta \end{bmatrix} \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix}_e \Rightarrow v = \mathbf{N}_v \mathbf{v}_e \quad (3.12)$$

where the shape functions are given by

$$\mathcal{N}_1^v = 1 - 3\left(\frac{x_\ell}{L}\right)^2 + 2\left(\frac{x_\ell}{L}\right)^3 \quad (3.13)$$

$$\mathcal{N}_1^\theta = L \left[\left(\frac{x_\ell}{L}\right) - 2\left(\frac{x_\ell}{L}\right)^2 + \left(\frac{x_\ell}{L}\right)^3 \right] \quad (3.14)$$

$$\mathcal{N}_2^v = 3\left(\frac{x_\ell}{L}\right)^2 - 2\left(\frac{x_\ell}{L}\right)^3 \quad (3.15)$$

$$\mathcal{N}_2^\theta = L \left[-\left(\frac{x_\ell}{L}\right)^2 + \left(\frac{x_\ell}{L}\right)^3 \right] \quad (3.16)$$

The shape functions are displayed in Figure 3.2 and have been determined to be consistent with all established criteria in Eq. (3.4) - (3.7).

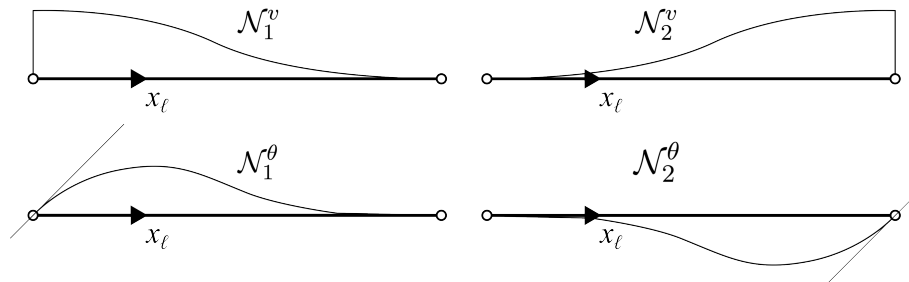


Figure 3.2: The beam elements' shape functions.

Stress and Strain

According to beam theory, the planar cross-sections of a beam remain planar after deformation or bending if they rotate through a small angle dv/dx (Krenk and Høgsberg, 2012). Subsequently, an expression for the curvature κ ,

$$\kappa = \frac{d^2v}{dx^2} \quad (3.17)$$

is obtained for the beam when the angles are considered to be of small order. The curvature is placed in the strain matrix ε , similar to Eq. (2.7). The strain vector is obtained by substituting Eq. (3.12) into Eq. (3.17),

$$\varepsilon = [\kappa] = \left[\frac{d}{dx^2} \right] \begin{bmatrix} \mathcal{N}_1^v & \mathcal{N}_1^\theta & \mathcal{N}_2^v & \mathcal{N}_2^\theta \end{bmatrix} \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix}_e \quad (3.18)$$

Thus, the strain interpolation matrix for the beam element $\mathbf{B}_v$ is introduced as

$$\mathbf{B}_v = \left[\frac{d}{dx^2} \right] \begin{bmatrix} \mathcal{N}_1^v & \mathcal{N}_1^\theta & \mathcal{N}_2^v & \mathcal{N}_2^\theta \end{bmatrix} \quad (3.19)$$

whereby, Eq. (3.18) can be formulated as

$$\varepsilon = \mathbf{B}_v \mathbf{v}_e \quad (3.20)$$

The bending moment, denoted by M , is the only section force acting in the 2D beam element and therefore included in the stress state vector

$$\boldsymbol{\sigma} = [M] = [EI] [\kappa] \quad (3.21)$$

where the material stiffness matrix

$$\mathbf{D} = [EI] \quad (3.22)$$

contains EI , the elastic bending stiffness. Hence, the stress state of the element is expressed in compact form

$$\boldsymbol{\sigma} = \mathbf{D} \varepsilon = \mathbf{D} \mathbf{B}_v \mathbf{v}_e \quad (3.23)$$

describing the generalized forces present during deformation.

3.2 Finite Element Formulation for Bending Vibration

The transverse motion of the beam is described by the differential equation

$$(EI v'')'' + \rho A \ddot{v} - q(x, t) = 0 \quad (3.24)$$

where the transverse vibration, $v(x, t)$ and transverse force, $q(x, t)$ are dependent on both time and x along the local longitudinal axis (Krenk, 2018).

3.2.1 The Principle of Virtual Work

The force equation Eq. (3.24) is multiplied by a virtual displacement and integrated over the length of the beam. The virtual work formulation is given by

$$\int_0^L \delta v \{ (EI v'')'' + \rho A \ddot{v} - q \} dx = 0 \quad (3.25)$$

The same method of derivation is applied as previously demonstrated in the case of axial motion in Section 2.2. The first term in the work formulation is integrated twice using the integration-by-parts method,

$$\int_0^L \delta v (EI v'')' dx = \int_0^L \delta v'' EI v'' dx + [\delta v (EI v'')' - \delta v' (EI v'')]_0^L \quad (3.26)$$

ensuring the first term in Eq. (3.25) becomes symmetrical. When Eq. (3.26) is substituted back into Eq. (3.25), the terms in the weak form equation are separated into

$$\int_0^L \delta v'' EI v'' dx + \int_0^L \delta v \rho A \ddot{v} dx = \int_0^L \delta v q dx + [\delta v' (EI v'') - \delta v (EI v'')]_0^L \quad (3.27)$$

in which the right hand side combines equivalent nodal loads and isolated nodal loads.

3.2.2 Implementing Shape Functions

The transverse vibration is reformulated using the definitions in Eq. (3.12) and Eq. (3.18). The same shape functions are employed as shown in Figure 3.2, where displacement and curvature are represented in vectorial form as

$$v(x, t) = \mathbf{N}_v \mathbf{v}_e = [\mathcal{N}_1^v \quad \mathcal{N}_1^\theta \quad \mathcal{N}_2^v \quad \mathcal{N}_2^\theta] \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix}_e \quad (3.28)$$

$$\varepsilon = \mathbf{N}_v'' \mathbf{v}_e = \mathbf{B}_v \mathbf{v}_e = [B_1^v \quad B_1^\theta \quad B_2^v \quad B_2^\theta] \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix}_e \quad (3.29)$$

It is noted that from the square bracket in Eq. (3.27), the shear force is defined by $Q = -(EI v'')'$ and the bending moment is defined by $M = -EI v''$. All relations are substituted into Eq. (3.27), which can be formulated as

$$\begin{aligned} & \int_0^L (\delta \mathbf{v}_e)^T \mathbf{B}_v^T EI \mathbf{B}_v \mathbf{v}_e dx + \int_0^L (\delta \mathbf{v}_e)^T \mathbf{N}_v^T \rho A \mathbf{N}_v \ddot{\mathbf{v}}_e dx \\ &= \int_0^L (\delta \mathbf{v}_e)^T \mathbf{N}_v^T q dx + [(\delta \mathbf{v}_e)^T \mathbf{N}_v^T Q - (\delta \mathbf{v}_e)^T (\mathbf{N}_v')^T M]_0^L \end{aligned} \quad (3.30)$$

using shape functions $\mathbf{N}_v$ to represent inertia and load, while the strain interpolation $\mathbf{B}_v$ distributes stiffness.

3.2.3 Deriving the Equation of Motion

The terms in Eq. (3.30) are arranged on the right-hand side of the equation in order to factorize by using the common factor $(\delta \mathbf{v}_e)^T$. Using the zero-product property, the equation of motion yields

$$\begin{aligned} & \int_0^L \mathbf{B}_v^T EI \mathbf{B}_v dx \mathbf{v}_e + \int_0^L \mathbf{N}_v^T \rho A \mathbf{N}_v dx \ddot{\mathbf{v}}_e \\ &= \int_0^L \mathbf{N}_v^T q dx + [\mathbf{N}_v^T Q - (\mathbf{N}_v')^T M]_0^L \end{aligned} \quad (3.31)$$

From Eq. (3.31) it follows that the element stiffness matrix, $\mathbf{k}_e$ and the element mass matrix, $\mathbf{m}_e$ can be expressed by the integrals

$$\mathbf{k}_e = \int_0^L \mathbf{B}_v^T EI \mathbf{B}_v dx = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (3.32)$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}_v^T \rho A \mathbf{N}_v dx = \frac{\rho AL}{420} \begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \quad (3.33)$$

which in the present case are evaluated analytically.

3.3 Assembling System Matrices for the 2D Beam

To establish the relationship between local and global coordinates, a transformation matrix $\mathbf{A}$ is introduced. The transverse displacement v_e in the local coordinate system can be expressed as

$$v_e = u(-\sin \alpha) + v \cos \alpha \quad (3.34)$$

where the direction cosines ($\cos \alpha$ and $\sin \alpha$) can be computed using Eq. (2.29), see also Figure 2.3. No transformation is required for the rotational degree of freedom θ , as for a rotation $\theta_{\text{Local}} = \theta_{\text{Global}}$ in 2D problems. Based on Eq. (3.34), the relationship between local and global nodal displacements is formulated in matrix form (Nielsen et al., 2019) as

$$\begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix}_e = \begin{bmatrix} -n_y & n_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -n_y & n_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ \theta_i \\ u_j \\ v_j \\ \theta_j \end{bmatrix} \Rightarrow \mathbf{v}_e = \mathbf{A} \mathbf{d} \quad (3.35)$$

Consequently, the same transformation relationships as presented in Eq. (2.38) are obtained. The global system matrices are assembled by the method described in Section 2.3.2.

3.4 2D Beam Element with Axial and Transverse Effects

In order to model a two-dimensional beam element capable of resisting axial, shear, and bending forces, the theories previously developed for the 2D bar element and 2D beam element are combined.

3.4.1 Combined Virtual Work

The axial element formulation outlined in Eq. (2.17) and the bending behavior described previously in Eq. (3.27) are summed into a single element formulation by

$$\begin{aligned} & \int_0^L \{ \delta u' EA u' + \delta v'' EI v'' \} dx + \int_0^L \{ \delta u \rho A \ddot{u} + \delta v \rho A \ddot{v} \} dx \\ & = [\delta u N + \delta v Q - \delta v' M]_0^L + \int_0^L \{ \delta u p + \delta v q \} dx \end{aligned} \quad (3.36)$$

In order to combine the terms in each integral the displacement interpolation matrix is introduced,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \mathcal{N}_1^u & 0 & 0 & \mathcal{N}_2^u & 0 & 0 \\ 0 & \mathcal{N}_1^v & \mathcal{N}_1^\theta & 0 & \mathcal{N}_2^v & \mathcal{N}_2^\theta \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{bmatrix}_e = \mathbf{N} \mathbf{d}_e \quad (3.37)$$

which interpolates both u and v simultaneously. Likewise, the strain state is expanded and the strain interpolation matrix of shape 2×6 is obtained by

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \kappa \end{bmatrix} = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d^2}{dx^2} \end{bmatrix} \begin{bmatrix} \mathcal{N}_1^u & 0 & 0 & \mathcal{N}_2^v & 0 & 0 \\ 0 & \mathcal{N}_1^v & \mathcal{N}_1^\theta & 0 & \mathcal{N}_2^v & \mathcal{N}_2^\theta \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{bmatrix}_e = \mathbf{B} \mathbf{d}_e \quad (3.38)$$

The material property matrix $\mathbf{D}$ is the link between the stress states (normal force N and bending moment M) and the strains, defined by

$$\boldsymbol{\sigma} = \begin{bmatrix} N \\ M \end{bmatrix} = \begin{bmatrix} EA & 0 \\ 0 & EI \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \kappa \end{bmatrix} = \mathbf{D} \boldsymbol{\varepsilon} \quad (3.39)$$

Substituting the coupled interpolation matrices from Eq. (3.38) and the material property matrix in Eq. (3.39) into Eq. (3.36) results in

$$\left(\int_0^L \mathbf{B}^T \mathbf{D} \mathbf{B} dx \right) \mathbf{d}_e + \left(\int_0^L \mathbf{N}^T \rho A \mathbf{N} dx \right) \ddot{\mathbf{d}}_e = \mathbf{f}_e \quad (3.40)$$

Finally, the element stiffness matrix $\mathbf{k}_e$ and mass matrix $\mathbf{m}_e$ are obtained by the integrals

$$\mathbf{k}_e = \int_0^L \mathbf{B}^T \mathbf{D} \mathbf{B} dx \quad (3.41)$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}^T \rho A \mathbf{N} dx \quad (3.42)$$

for a beam element with 6 DOFs, accounting for axial and transverse effects.

3.4.2 Assembling the System Matrix

The system matrices are assembled by transforming the element matrices from local to global coordinates as described in Eq. (2.37). The combined transformation matrix, $\mathbf{A}$ is introduced (Nielsen et al., 2019) by

$$\begin{bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{bmatrix}_e = \begin{bmatrix} n_x & n_y & 0 & 0 & 0 & 0 \\ -n_y & n_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & n_x & n_y & 0 \\ 0 & 0 & 0 & -n_y & n_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ \theta_i \\ u_j \\ v_j \\ \theta_j \end{bmatrix} \Rightarrow \mathbf{d}_e = \mathbf{A} \mathbf{d} \quad (3.43)$$

The elements matrices in global coordinates are assembled into the system matrices, as outlined in Section 2.3.2. The indices used to define the element system matrices $\mathbf{K}_{ele}$ and $\mathbf{M}_{ele}$ are defined by Eq. (2.54), as the 2D beam element contains a total of six DOFs when the axial effects are included.

3.5 The 3D Beam Element

Finally, the 3D beam element is introduced in Figure 3.3, where torsional effects are considered. The element is modeled with two rotational DOFs ϕ , corresponding to the angle of twist at each node. When modeling torsional vibration in 3D beams, it is generally appropriate to replace the polar moment of area J with the torsional constant γ for non-circular sections to account for shear deformation and warping effects. This allows for a more accurate representation of torsional stiffness and dynamic response (Inman, 2007).

The following theory will be introduced using γ . In the application of the beam element, it is utilized that $\gamma = J$ for rods. The local nodal twist vector and the corresponding nodal load vector are defined as

$$\phi_e = [\phi_1 \quad \phi_2]_e^T \quad (3.44)$$

$$\mathbf{s}_e = [s_1 \quad s_2]_e^T \quad (3.45)$$

where s corresponds to a torsional nodal moment.

Shape Functions

A linear angle-of-twist variation is assumed throughout the beam element. The interpolation of the twist angle $\phi(x_\ell)$ is given by

$$\phi = [\mathcal{N}_1^\phi \quad \mathcal{N}_2^\phi] \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}_e \Rightarrow \phi = \mathbf{N}_\phi \phi_e \quad (3.46)$$

where the shape functions $\mathcal{N}_1^\phi$ and $\mathcal{N}_2^\phi$ are equivalent to the shape functions $\mathcal{N}_1^u$ and $\mathcal{N}_2^u$ from Eq. (2.3).

Stress and Strain

The relevant strain measure is the rate of twist ϕ' , which represents the gradient of the twist angle along the element axis. Differentiating Eq. (3.46) with respect to the local coordinate x_ℓ yields

$$\phi' = \mathbf{B}_\phi \phi_e \quad (3.47)$$

where the strain interpolation matrix $\mathbf{B}_\phi$ is defined as

$$\mathbf{B}_\phi = \frac{d}{dx} [\mathcal{N}_1^\phi \quad \mathcal{N}_2^\phi] \quad (3.48)$$

The internal stress quantity associated with torsion is the torsional moment M_x , which is linearly related to the rate of twist by

$$\sigma = [M_x] = [G\gamma] \phi' = \mathbf{D} \mathbf{B}_\phi \phi_e \quad (3.49)$$

where $G\gamma$ is as the torsional stiffness.

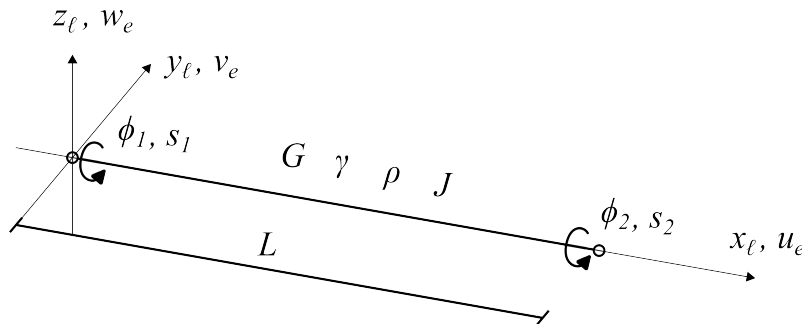


Figure 3.3: Definition of the 3D beam element (torsion).

3.6 Finite Element Formulation for Torsional Vibration

The differential equation describing torsional vibration $\phi(x, t)$ is defined by

$$(G\gamma\phi')' - \rho J\ddot{\phi} + s = 0 \quad (3.50)$$

where ρJ is the rotational mass moment of inertia and $s(x, t)$ is the distributed torsional moment (Inman, 2007).

3.6.1 The Principle of Virtual Work

The finite element formulation is derived by the method applied for axial motion in Section 2.2. The work equation is obtained

$$\int_0^L \delta\phi \left\{ (G\gamma\phi')' - \rho J\ddot{\phi} + s \right\} dx = 0 \quad (3.51)$$

by multiplying a virtual torsion $\delta\phi$ to Eq. (3.50) and integrating over the length of the element. Employing partial integration will ensure that the first term in equation Eq. (3.51) becomes symmetrical. The weak form equation is given by

$$\int_0^L \delta\phi' G\gamma\phi' dx + \int_0^L \delta\phi \rho J \ddot{\phi} dx = [\delta\phi (G\gamma\phi')]_0^L + \int_0^L \delta\phi s dx \quad (3.52)$$

where the right hand side combines equivalent nodal moments and isolated nodal moments.

3.6.2 Implementing Shape Functions

Subsequently, the relations Eq. (3.46) and Eq. (3.47), are substituted into Eq. (3.52) and the equation of motion is obtained

$$\left(\int_0^L \mathbf{B}_\phi^T G\gamma \mathbf{B}_\phi dx \right) \phi_e + \left(\int_0^L \mathbf{N}_\phi^T \rho J \mathbf{N}_\phi dx \right) \ddot{\phi}_e = [\mathbf{N}_\phi^T M_x]_0^L + \int_0^L \mathbf{N}_\phi^T s dx \quad (3.53)$$

employing that the virtual rotation $(\delta\phi_e)^T \neq 0$. The element stiffness and mass matrices are defined by the integrals

$$\mathbf{k}_e = \int_0^L \mathbf{B}_\phi^T G\gamma \mathbf{B}_\phi dx = \frac{G\gamma}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3.54)$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}_\phi^T \rho J \mathbf{N}_\phi dx = \frac{\rho J L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (3.55)$$

which are evaluated analytically.

3.7 Assembling System Matrices for the 3D Beam

Consider torsion, ϕ to be a vector pointing along the x_ℓ axis. The transformation of the vector between local and global coordinates is identical to the transformation previously described in Section 2.5.1. Therefore, the element transformation matrix is expressed by

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}_e = \begin{bmatrix} \cos \alpha_x & \cos \alpha_y & \cos \alpha_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha_x & \cos \alpha_y & \cos \alpha_z \end{bmatrix} \begin{bmatrix} \phi_{u1} \\ \phi_{v1} \\ \phi_{w1} \\ \phi_{u2} \\ \phi_{v2} \\ \phi_{w2} \end{bmatrix} \Rightarrow \phi_e = \mathbf{A} \phi \quad (3.56)$$

The same applies to the warping moments in $\mathbf{s}_e$.

3.8 3D Beam Element with Axial, Transverse and Torsional Effects

This section presents the derivation of a 3D beam element, accounting for axial deformation, bending in both transverse directions, and torsional effects. The beam is assumed to have six degrees of freedom per node: three translational (u, v, w) and three rotational ($\phi_x, \theta_y, \theta_z$).

3.8.1 Combined Virtual Work

Axial Deformation

The contribution from axial deformation is given by Eq. (2.17).

Transverse Bending in the x - y Plane

The transverse vibration for bending in the x - y plane is described by

$$\int_0^L \delta v'' EI_z v'' dx + \int_0^L \delta v \rho A \ddot{v} dx = [\delta v Q_y - \delta v' M_z]_0^L + \int_0^L \delta v q dx \quad (3.57)$$

which is associated with rotation about the z -axis, denoted by θ_z .

Transverse Bending in the x - z Plane

The transverse vibration for bending in the x - z plane is described by

$$\int_0^L \delta w'' EI_y w'' dx + \int_0^L \delta w \rho A \ddot{w} dx = [\delta w Q_z - \delta w' M_y]_0^L + \int_0^L \delta w r dx \quad (3.58)$$

which is associated with rotation about the y -axis, denoted by θ_y .

Torsional Deformation

The torsional contribution, which includes both strain energy due to twisting and torsional inertia, is expressed in Eq. (3.52). The total internal virtual work for the combined element is the sum of the axial, bending, and torsional contributions

$$\delta W_k = \int_0^L \delta u' EA u' + \delta v'' EI_z v'' + \delta w'' EI_y w'' + \delta \phi' G \gamma \phi' dx \quad (3.59)$$

which in this demonstration only includes the stiffness-related terms. A similar formulation holds for the inertial contributions following the same approach, which is not demonstrated here. To represent the displacement field along the beam, the displacement interpolation matrix is introduced,

$$\begin{bmatrix} u \\ v \\ w \\ \phi \end{bmatrix} = \begin{bmatrix} \mathcal{N}_1^u & 0 & 0 & 0 & 0 & 0 & \mathcal{N}_2^u & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{N}_1^v & 0 & 0 & 0 & \mathcal{N}_1^{\theta_z} & 0 & \mathcal{N}_2^v & 0 & 0 & 0 & \mathcal{N}_2^{\theta_z} \\ 0 & 0 & \mathcal{N}_1^w & 0 & -\mathcal{N}_1^{\theta_y} & 0 & 0 & 0 & \mathcal{N}_2^w & 0 & -\mathcal{N}_2^{\theta_y} & 0 \\ 0 & 0 & 0 & \mathcal{N}_1^\phi & 0 & 0 & 0 & 0 & 0 & \mathcal{N}_2^\phi & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \varphi_1 \\ \theta_{1y} \\ \theta_{1z} \\ u_2 \\ v_2 \\ w_2 \\ \varphi_2 \\ \theta_{2y} \\ \theta_{2z} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u \\ v \\ w \\ \phi \end{bmatrix} = \mathbf{N} \mathbf{d}_e \quad (3.60)$$

which interpolates both u, v, w and ϕ simultaneously. The generalized strain vector is determined by the expression

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \kappa_z \\ \kappa_y \\ \phi' \end{bmatrix} = \begin{bmatrix} \frac{d}{dx} & 0 & 0 & 0 \\ 0 & \frac{d^2}{dx^2} & 0 & 0 \\ 0 & 0 & \frac{d^2}{dx^2} & 0 \\ 0 & 0 & 0 & \frac{d}{dx} \end{bmatrix} \mathbf{N} \mathbf{d}_e = \mathbf{B} \mathbf{d}_e \quad (3.61)$$

Furthermore the material property matrices are expanded in alignment with Eq. (3.59) as

$$\mathbf{D}_k = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & EI_z & 0 & 0 \\ 0 & 0 & EI_y & 0 \\ 0 & 0 & 0 & GJ \end{bmatrix}, \quad \mathbf{D}_m = \begin{bmatrix} \rho A & 0 & 0 & 0 \\ 0 & \rho A & 0 & 0 \\ 0 & 0 & \rho A & 0 \\ 0 & 0 & 0 & \rho J \end{bmatrix} \quad (3.62)$$

and the stress state is expanded using,

$$\boldsymbol{\sigma} = \begin{bmatrix} N_x \\ M_z \\ M_y \\ M_x \end{bmatrix} = \mathbf{D}_k \boldsymbol{\varepsilon} \quad (3.63)$$

which concludes the definitions of interpolation and material property matrices for the combined 3D element.

Global Equation of Motion

When combining the coupled interpolation matrices, the global finite element equation ultimately leads to

$$\left(\int_0^L \mathbf{B}^T \mathbf{D}_k \mathbf{B} dx \right) \mathbf{d}_e + \left(\int_0^L \mathbf{N}^T \mathbf{D}_m \mathbf{N} dx \right) \ddot{\mathbf{d}}_e = \mathbf{f}_e \quad (3.64)$$

where $\mathbf{f}$ contains all external nodal loads and moments.

$$\mathbf{f}_e = [p_1 \quad q_1 \quad r_1 \quad m_{1x} \quad m_{1y} \quad m_{1z} \quad p_2 \quad q_2 \quad r_2 \quad m_{2x} \quad m_{2y} \quad m_{2z}]^T \quad (3.65)$$

Finally, the element stiffness matrix $\mathbf{k}_e$ and mass matrix $\mathbf{m}_e$ for a beam element with 12 DOFs in 3D are obtained by the integrals

$$\mathbf{k}_e = \int_0^L \mathbf{B}^T \mathbf{D}_k \mathbf{B} dx \quad (3.66)$$

$$\mathbf{k}_e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix}$$

$$\mathbf{m}_e = \int_0^L \mathbf{N}^T \mathbf{D}_m \mathbf{N} dx \quad (3.67)$$

$$\mathbf{m}_e = \frac{L}{420} \begin{bmatrix} 140\rho A & 0 & 0 & 0 & 0 & 0 & 70\rho A & 0 & 0 & 0 & 0 & 0 \\ 0 & 156\rho A & 0 & 0 & 0 & 22L\rho A & 0 & 54\rho A & 0 & 0 & 0 & -13L\rho A \\ 0 & 0 & 156\rho A & 0 & -22L\rho A & 0 & 0 & 0 & 54\rho A & 0 & 13L\rho A & 0 \\ 0 & 0 & 0 & 140\rho J & 0 & 0 & 0 & 0 & 0 & 70\rho J & 0 & 0 \\ 0 & 0 & -22L\rho A & 0 & 4L^2\rho A & 0 & 0 & 0 & -13L\rho A & 0 & -3L^2\rho A & 0 \\ 0 & 22L\rho A & 0 & 0 & 0 & 4L^2\rho A & 0 & 13L\rho A & 0 & 0 & 0 & -3L^2\rho A \\ 70\rho A & 0 & 0 & 0 & 0 & 0 & 140\rho A & 0 & 0 & 0 & 0 & 0 \\ 0 & 54\rho A & 0 & 0 & 0 & 13L\rho A & 0 & 156\rho A & 0 & 0 & 0 & -22L\rho A \\ 0 & 0 & 54\rho A & 0 & -13L\rho A & 0 & 0 & 0 & 156\rho A & 0 & 22L\rho A & 0 \\ 0 & 0 & 0 & 70\rho J & 0 & 0 & 0 & 0 & 0 & 140\rho J & 0 & 0 \\ 0 & 0 & 13L\rho A & 0 & -3L^2\rho A & 0 & 0 & 0 & 22L\rho A & 0 & 4L^2\rho A & 0 \\ 0 & -13L\rho A & 0 & 0 & 0 & -3L^2\rho A & 0 & -22L\rho A & 0 & 0 & 0 & 4L^2\rho A \end{bmatrix}$$

Figure 3.4 shows the configuration of a 3D beam element, which consists of 12 degrees of freedom, allowing for translation and rotation in all three spatial directions.

3.8.2 Assembling the System Matrix

The system matrices are assembled by transforming the element matrices from local to global coordinates as outlined in Eq. (2.37). For the complete element transformation in 3D, a block-diagonal transformation matrix $\mathbf{A}$ applied. Each diagonal block is a 3x3 rotation matrix T , defined by Eq. (2.52), that acts on either translations or rotations

$$\mathbf{A} = \begin{bmatrix} T_{3 \times 3} & & & \\ & T_{3 \times 3} & & \\ & & T_{3 \times 3} & \\ & & & T_{3 \times 3} \end{bmatrix} \quad (3.68)$$

where the first two blocks correspond to rotations translations at node i and the last two to those at node j . The corresponding relationships between the global and local element quantities are given by Eq. (2.38). The elements matrices in global coordinates are combined into the system matrices, as outlined in Section 2.3.2 with the index array

$$\mathbf{i}_{ele} = [6i - 5 \quad 6i - 4 \quad 6i - 3 \quad 6i - 2 \quad 6i - 1 \quad 6i \quad 6j - 5 \quad 6j - 4 \quad 6j - 3 \quad 6i - 2 \quad 6j - 1 \quad 6j]_{ele} \quad (3.69)$$

which is defined for an element with a total of 12 DOFs.

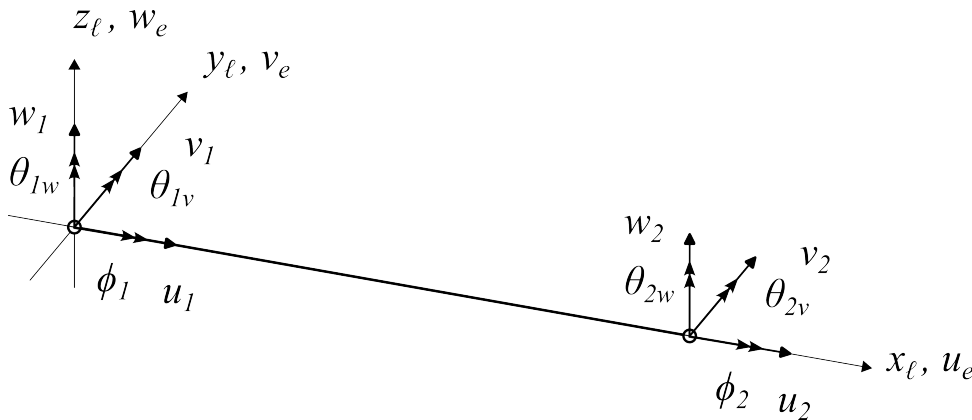


Figure 3.4: The 3D beam element with 12 DOFs.

3.9 Mode Shapes and Natural Frequencies

The objective of the FE program is to perform a Natural Frequency Analysis (NFA). Natural frequencies are defined as the frequencies of a free-vibration system, i.e. a system with no force or damping. Consequently, the natural frequencies are determined exclusively by the characteristics of the system, namely its stiffness and mass (Krenk, 2018). The response of the free-vibration system is described by the equation of motion

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{0} \quad (3.70)$$

where $\mathbf{M}$, $\mathbf{K}$, and $\mathbf{d}$ refer to the global system matrices and vectors. The solution to the system of differential equations is given by the harmonic equation

$$\mathbf{d}(t) = \mathbf{u} e^{i\omega t} \quad (3.71)$$

where ω is the natural angular frequency, and the vector $\mathbf{u}$ represents the nodal displacements known as the natural modes. By differentiating Eq. (3.71) twice with respect to time, it is obtained that

$$\ddot{\mathbf{d}}(t) = -\omega^2 \mathbf{u} e^{i\omega t} \quad (3.72)$$

Eq. (3.72) and Eq. (3.71) are substituted into Eq. (3.70) and factorized by $e^{i\omega t}$

$$\Rightarrow -\omega^2 \mathbf{M} \mathbf{u} e^{i\omega t} + \mathbf{K} \mathbf{u} e^{i\omega t} = e^{i\omega t} (-\omega^2 \mathbf{M} + \mathbf{K}) \mathbf{u} = \mathbf{0} \quad (3.73)$$

Since, $e^{i\omega t} \neq 0$, the substitution of the solution yields the generalized eigenvalue problem

$$(\mathbf{K} - \omega^2 \mathbf{M}) \mathbf{u} = \mathbf{0} \quad (3.74)$$

The nontrivial solution to the mode shapes $\mathbf{u}$ can only be found when ω^2 is an eigenvalue of the system. For a nontrivial solution, the determinant is zero and the eigenvalue is determined by the characteristic equation

$$\det(\mathbf{K} - \omega^2 \mathbf{M}) = 0 \quad (3.75)$$

The natural frequency is then determined by the square root of the eigenvalue, and the corresponding mode shape to the eigenvalue is determined by solving the linear system. The complete solution to the eigenvalue problem includes all quantities of eigenvalues and corresponding eigenvectors. The number of unbounded degrees of freedom determines the number of solutions. The definition and assembly of the system mass and stiffness matrices lead to real-symmetric matrix properties.

4 Development of the FE program

The theory presented in the former chapters provide the foundation for the development of the FE programs. A program is developed on the basis of each of the presented elements: the 2D bar, the 2D beam, the 3D bar and the 3D beam. The following chapter outlines the main objectives for the programs, the approach taken in implementation and the rationale behind some of the key decisions.

4.1 Objective and Application

The main objective of the project is to develop an in-house program for natural frequency and mode shape analysis. The programs are developed to be cost-effective, minimizing both the time users spend on application and execution. Specifically designed for the jacket structure, the tool should offer flexibility through a partially parametric implementation. This allows users to modify the input data for analysis, enabling the investigation of various design choices.

The analysis in this project will focus on evaluating the accuracy of simple FE models. All four developed programs provide simplified solutions for computing natural frequencies and mode shapes compared to well-established FE programs, commonly used in wind turbine design. The developed programs are designed with a 'plug and play' approach, ensuring accessibility and facilitating a more efficient solution process. Developed in Python, the programs demand only basic understanding of the syntax. Additionally, Python is an open-source software, ensuring unrestricted access for the users.

4.2 Program Structure

All four Python programs have the same base structure, visualized in Figure 4.1. Each program's overall framework is divided into two sets of files: User-controlled Features and Adaptions, and Structural Framework.

4.2.1 Structural Framework

The Structural Framework consists of functions for executing the core analytical processes, which is not intended for user modification. This part of the program is responsible for the numerical computation of natural frequencies and mode shapes, data storage operations, and plotting. The program is organized in six directories, each containing functions dedicated to a specific aspect of operation.

Data Storing and Output

Within the data directory lies the functions for the collection of data and storing results. The functions include `baseclear`, `baseinsert` and `basestore`. The function `basestore` initializes a database and calls the function `baseclear`, which clears and overwrites an existing database. The `basestore` function then calls `baseinsert`, which is responsible for inserting the results in the database.

The directory also contains the function `output`, which creates a `.txt` file with results. The output file retrieves the natural frequencies directly from a data object and lists the natural frequencies and corresponding mode numbers.

Building Input Data

The functions within the input data directory process user inputs and generate input data for the analysis of the jacket structure. The function `buildx` creates the node coordinate matrix for the jacket structure. In order to define the node coordinates, helper functions in `geometry` are used to

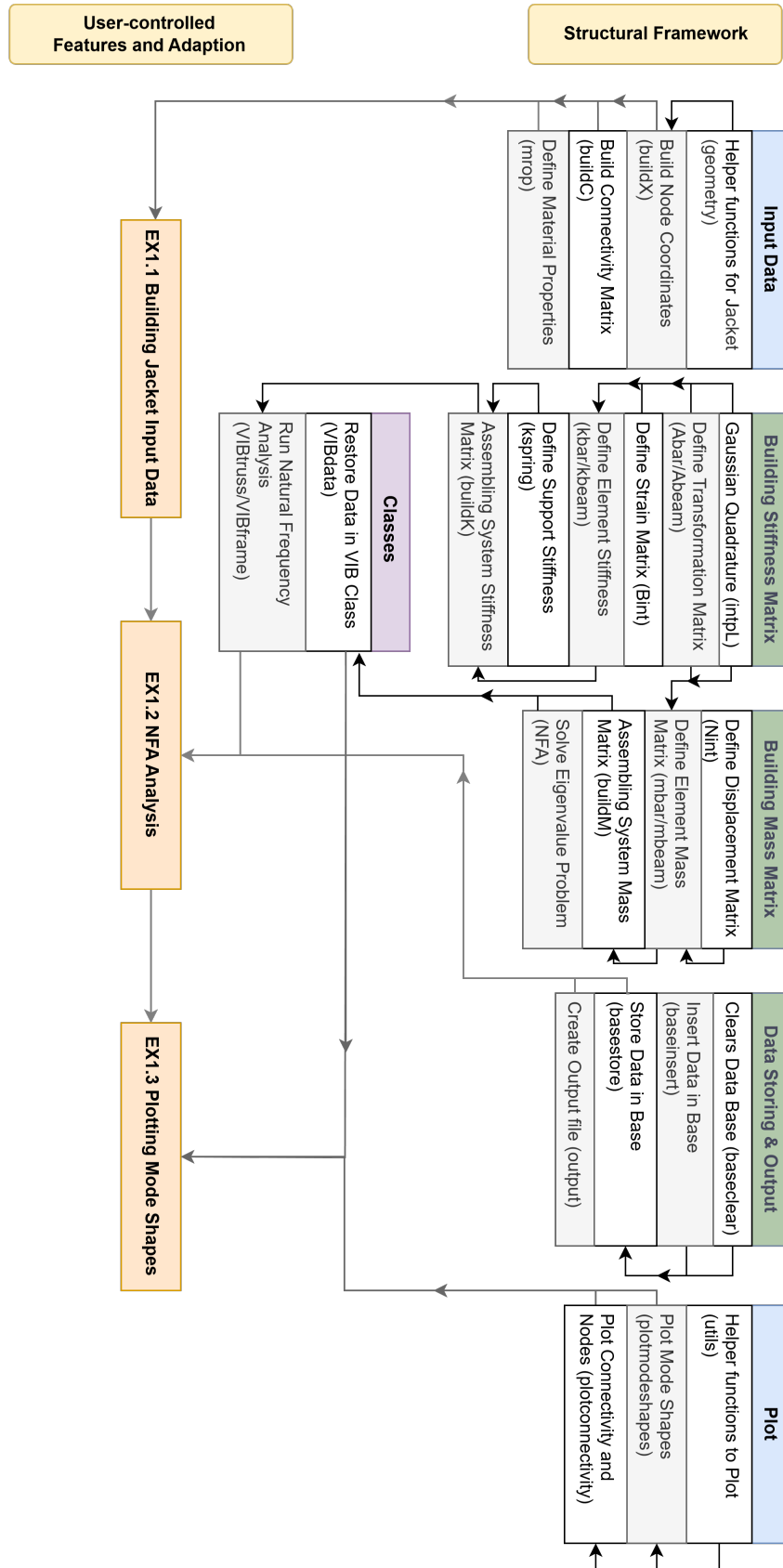


Figure 4.1: Flow chart over program structure. Each table represents a folder and each row a function file. User-controlled features are executed in separate files, which can be altered.

**Code 4.1: VIBframe**

```

class VIBframe():
    def __init__(self, X, C, mprop, bound, spring_support, solve_subset =
None): ...
    # Assign input to object and initialize
    ...
    # Build the mass matrix, M
    def buildM(self):
        self.M = buildM(self.X, self.C, self.mprop, self.nno, self.nne,
self.ldof)
    # Build the system stiffness matrix, K
    def buildK(self):
        self.K = buildK(self.X, self.C, self.mprop, self.spring_support,
self.nno, self.nne, self.ldof)
    # Solve the eigenvalue problem
    def NFA(self):
        self.omega, self.U = NFA(self.K, self.M, self.nno, self.ldof,
self.bound, self.solve_subset)

```

determine angles and points of intersection. In buildC the connectivity matrix is defined. Within the connectivity matrix a number format is used as a reference to a material library constructed by mprop.

The 2D model represents a single side of the jacket structure, created as a planar projection. The 3D model corresponds to a four-legged jacket structure, defined according to the properties in Chapter 1, with user-defined inputs detailed in Section 4.2.2.

Classes

As shown in Figure 4.1, the directory for constructing system matrices and running the NFA does not directly retrieve data from the input data directory. Instead, the VIBtruss/VIBframe class governs the analysis, as shown in Code 4.2.1. The class environment takes the defined input, execute functions to construct the system stiffness and system mass matrix, and finally solves the eigenvalue problem.

The VIBdata class works by taking in a database as input. The database is created prior by the basestore function. The VIBdata class then creates an object, similar to VIBtruss/VIBframe, which contains the data needed for creating output and plotting. Differently to the VIBtruss/VIBframe class, VIBdata has a set of functions, which are defined directly in the class rather than in a separate directory. These functions are created to pull the designated data from the database.

Building the System Stiffness Matrix

The system matrices are built on the basis of the finite element formulations. Initially, the the strain interpolation matrix is defined. For the bar element it is not dependent on the local coordinate, and can therefore be determined by matrix product of strain interpolation matrix and material-stiffness matrix as according to Eq. (2.25), which is executed in kbar.

For the beam element the strain interpolation functions are dependent on the local coordinate, which is defined by the function Bint. Therefore, Eq. (3.32) is integrated using Gauss-Legendre quadrature from intpL, described by Logan (2021). The structure of building the element stiffness matrix, is visualized in Figure 4.2. For both the truss and frame model, the transformation matrix is needed, which is computed in Abar/Abeam and retrieved in kbar/kbeam. The element stiffness matrices are retrieved by the function buildK, which builds the system stiffness matrix accord-

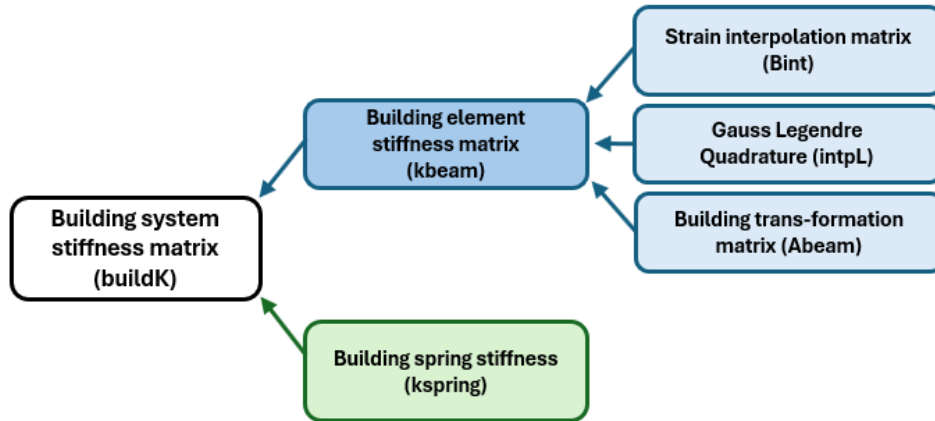


Figure 4.2: Process of building the beam model stiffness matrix.

ing to Eq. (2.42). Spring-stiffness is an optional feature, where the function `kspring` places the spring stiffness in a system spring matrix. The spring stiffness matrix is then retrieved by `buildK` and added to the system stiffness matrix. This is primarily thought as an alternative to defining boundary conditions. The user input for supports is described by Section 4.2.2.

Building the System Mass Matrix

The directory for constructing the system mass matrix follows the same process as the system stiffness matrix. The displacement interpolation matrix is defined in `Nint`. All element mass matrices are integrated using `intpL`. For all programs, the transformation matrix `Abar/Abeam` is retrieved from the system stiffness directory. The process is executed by the `mbar/mbeam` function, which is called by `buildM`.

Additionally, the directory for constructing the system mass matrix contains the `NFA` function for solving the eigenvalue problem. This function utilizes `eigh` from `scipy.linalg`, which is applicable as the matrices examined in the problem are real-symmetric (The SciPy Community, 2025). The structure of the `NFA` function allows to solve for a subset of modes, which confines the eigenvalue problem, thereby allowing faster computation.

Plotting Mode Shapes

The final directory handles visualization, featuring two plotting functions; `plotconnectivity`, which plots the jacket's geometry, and `plotmodeshapes`, which plots the obtained mode shapes. These functions rely on a collection of plotting utilities defined in `utils`.

4.2.2 User-controlled Features and Adaption

The User-Controlled Features and Adaption are demonstrated in a separate set of example files. These files can be modified to specific user needs. The natural frequency analysis is structured into three files for: (1) constructing the jacket input data, (2) executing the analysis and generating the output file, and (3) visualizing the mode shapes.

EX1.1: Building the Jacket Input Data

As described in Chapter 1, the project considers the design of a four-legged jacket structure with X-braces. The majority of parametric user-controlled properties are implemented during the generation of input data. This allows users to to modify, test, and design various jacket configurations. The design is limited to four-sided jackets with X-braces, using hollow rod members defined by outer diameter and wall thickness. At a given level, a single cross-section can be assigned to both brace and leg elements. The parametric choices for the analysis are presented in Section 6.1.1.

Table 4.1: User-controlled data parameters.

Parameter	Type	Unit
Width at mudline	float	m
Width at top	float	m
Height of jacket	float	m
Number of X-braces	int	-
Brace dimensions	numpy.array	m
Leg dimensions	numpy.array	m
Material elastic modulus	float	Pa
Material density	float	kg/m ³
Number of elements per beam	int	-

In the frame programs, beam elements can be subdivided to improve the numerical accuracy of the natural frequency estimation. However, this implementation is not included for the truss models due to the limitations of bar elements, which do not carry moments or transverse forces.

Additionally, an issue arises at the mid-brace connection in the 3D Truss model. As bar elements possess no transverse stiffness, the midpoint of the X-braces provides no structural support in the transverse axis. As a result, the mid-node of the X-braces has been removed from the 3D Truss model.

EX1.2 Execution of Natural Frequency

The second file intended for user adaptation performs an analysis for the input data independently from the data-generation process. Thereby, the natural frequency analysis (NFA) can be performed without the need to reconstruct the input data for each new set of conditions. Two methods are available for defining support conditions for the jacket structure. For both boundary conditions and spring supports, users must specify the node number (`node`) and the associated local DOF (`ldof`) where the support is applied. Boundary conditions are additionally defined by a prescribed displacement (`disp`), usually initialized to zero. For spring supports the constraint is defined by a spring stiffness (`stiffness`).

Additionally, users can define the input parameter `solve_subset` for VIBtruss/VIBframe to specify the range of mode numbers for which the natural frequency analysis (NFA) should be performed.

The VIBtruss/VIBframe classes create objects, as shown in Code 4.2.1, from which key results can be accessed, including; the system stiffness matrix K , the mass matrix M , the natural cyclic frequencies ω , and the corresponding mode shapes U . Following the analysis, the `basestore` function can be used to save both input and output data.

The output file is generated by extracting the VIBtruss/VIBframe object and creating a text file containing computed natural frequencies and corresponding mode numbers, which can be reviewed before plotting mode shapes.

Table 4.2: User options for support types.

Boundary condition	Spring support
numpy.array	numpy.array
[node, ldof, disp]	[node, ldof, stiffness]

EX1.3. Plotting Mode Shapes

The last file intended for user modification handles loading the data object via `VIBdata`. The `plotmodeshapes` function extracts the data object, takes an input for the mode number, and plots the corresponding mode shape.

4.3 Expansion of the Project Scope

During the process of development, the 2D Truss was the initial model, serving as the foundation for the structural framework of the 3D Truss, 2D Frame, and 3D Frame models. Following the implementation of the jacket, the scope was extended to include a transition piece, tower and RNA. These components are hard-coded. Therefore, modifying their values requires changes within the structural framework. The specific values are given in Section 6.1.1.

4.3.1 Transition piece

In all four program variants, the transition piece is modeled as a bar/beam element between the top nodes of the jacket.

4.3.2 Tower and RNA

The tower introduces modeling challenges, as bar elements cannot capture bending behavior. Consequently, the tower and RNA are implemented differently in the truss and frame programs.

Frame implementation

Both the 2D and 3D Frame models incorporate the tower directly into the node coordinates and connectivity matrices. When included, tower materials are hard-coded into `mprop`. The RNA is modeled as a lumped mass and rotational inertia applied at the tower's top node.

Truss implementation

In 2D and 3D Truss, the tower is implemented using beam elements, constructing a separate stiffness and mass matrix for the tower-RNA system. These matrices are imported from the corresponding frame models. The jacket and tower matrices are then coupled using Lagrange's method of undetermined multipliers, described in Cook et al. (1989), with implementation details provided in Appendices A.1 and A.2.

Due to the presence of Lagrange multipliers, the combined system matrices are non-symmetric. Therefore, the generalized eigenvalue solver `scipy.linalg.eig` is utilized (The SciPy Community, 2025). As a result, the computed eigenvalues must be filtered and sorted before computing the natural frequencies.

5 Program Validation and Testing

Having established the theoretical foundation and implemented the finite element formulations across all four programs, the subsequent step involves testing and validating these programs. This chapter presents a systematic validation of key components of the finite element formulation: the stiffness matrix, the mass matrix, the transformation matrix, and the resulting natural frequencies. A comparison between numerical and analytical solutions is crucial to ensure that the numerical models behave as expected under well-defined conditions, and provide a reliable basis for further analysis and application.

5.1 2D Truss

The foundation for 2D Truss validation is the free-free bar element in Figure 5.1. The bar element is assumed to have a constant cross-section.

5.1.1 The Stiffness Matrix

In Section 2.2, the 2D Truss equations have been developed, and the stiffness matrix $\mathbf{k}_e$ has been derived for the bar element using displacement functions, shown in Eq. (2.32). It should be noted that the strain matrix in 2D Truss is independent of x , which is exploited by

$$\mathbf{k}_e = \int_0^L \mathbf{B}^T \mathbf{D} \mathbf{B} dx = (\mathbf{B}^T \mathbf{D} \mathbf{B}) L \quad (5.1)$$

It can be demonstrated that, for the given element in Figure 5.1, the following equation holds: $\mathbf{K}_{sys} = \mathbf{k}_e$, since $\mathbf{A} = \mathbf{I}$.

The 2D Truss program is a numerical tool, which requires the assumption of values for A , E , and L to facilitate a comparison of the results obtained using Code 5.1 with the analytical solution in Eq. (2.32). The parameters of the bar in Table 5.1 are in accordance with the jacket's leg in Table 6.1 for geometric parameters, and Table 6.2 for material parameters.

Given that all elements in the stiffness matrix are dependent on E , A , and L , a unified solution will be presented by calculating $\mathbf{K}_{sys} L / (EA)$. The unified solution in Code 5.2 is obtained from the 2D Truss program after a subsequent filtration, that exclusively retains the axial components.

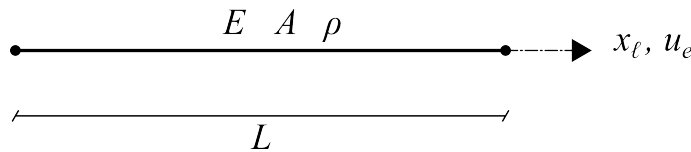


Figure 5.1: Free-free bar element.



Code 5.1: 2D Truss - Stiffness matrix implementation

```
A, L = Abar(n1,n2)
B = np.array([[ -1/L, 0, 1/L, 0]])
D = np.array([[Ge[0]*Ge[1]]])
# Define local stiffness matrix
k = B.T @ D @ B * L
```

Table 5.1: Material properties - free-free bar.

Element	A	E	L	ρ
Level A - Leg	$556.3 \cdot 10^{-3} \text{ m}^2$	210.0 GPa	19.86 m	8500 kg/m ³



Code 5.2: 2D Truss - Unified longitudinal stiffness matrix

```
# K_uni = K * L / (E*A)
K_uni = [[ 1. -1.]
         [-1.  1.]]
```

The accuracy of the numerically assembled stiffness matrix is evaluated by comparing it to the analytical formulation using

$$\Delta \mathbf{K}_{sys} = \mathbf{K}_{sys,analytical} - \mathbf{K}_{sys,numerical} \quad (5.2)$$

where $\Delta \mathbf{K}_{sys}$ is the error matrix. To ultimately evaluate the accuracy of the implementation, the Frobenius norm of $\Delta \mathbf{K}_{sys}$ is calculated in Python, using `numpy.linalg.norm` (NumPy Developers, 2025). This norm gives a single scalar value representing the magnitude of the difference between the two matrices. The relative error is calculated by

$$\mathbf{K}_{error} = \frac{\|\Delta \mathbf{K}_{sys}\|}{\|\Delta \mathbf{K}_{sys,analytical}\|} = 2.220 \cdot 10^{-16} \sim 0 \quad (5.3)$$

A comparison between the results in Code 5.2 and Eq. (2.25) indicates that the numerical and analytical stiffness matrices are well aligned with the chosen implementation.

5.1.2 The Mass Matrix

In Section 2.2 the 2D consistent-mass matrix $\mathbf{m}_e$ has been derived for the bar element, see Eq. (2.26). The shape functions and consequently, the matrix $\mathbf{N}$ are dependent on x . In this case, all terms in Eq. (2.26) must be integrated. In implementing the programs, the decision has been made to employ the isoparametric formulation for the bar and the Gaussian quadrature method to solve Eq. (2.31), see Code 5.3. The method is described in more detail in Logan (2021). Thereby Eq. (2.31) adapted to

$$\mathbf{m}_e = \rho A \int_0^L \mathbf{N}^T \mathbf{N} dx = \rho A |J| \int_{-1}^1 \mathbf{N}_{iso}^T \mathbf{N}_{iso} ds \quad (5.4)$$



Code 5.3: 2D Truss - Mass matrix implementation

```
A, L = Abar(n1,n2)
xip, wip = intpL(PolDeg)
J = L/2
# Define local mass matrix using Gaussian quadrature
m_ = np.zeros((4,4))
for i in range(len(xip)):
    N = Nint(xip[i])
    m_ += N.T @ N * wip[i] * J * Ge[0] * Ge[1]
```



Code 5.4: 2D Truss - Unified longitudinal mass matrix

```
# M_uni = M * 6/(rho*A*L)
M_uni = [[ 2.  1.]
         [ 1.  2.]]
```

In order to facilitate a comparison of the results obtained using Code 5.3 with the analytical solution in Eq. (2.26), the same system parameters from Table 5.1 are employed. The unified solution in Code 5.4 is obtained by calculating $\mathbf{M}_{sys} 6/(\rho AL)$, which exclusively retains the axial components.

The numerical mass matrix is validated by comparing it against the analytical expression

$$\Delta \mathbf{M}_{sys} = \mathbf{M}_{sys,analytical} - \mathbf{M}_{sys,numerical} \quad (5.5)$$

where $\Delta \mathbf{M}_{sys}$ is the error matrix. The norm is used to represent the magnitude of the difference between the two matrices

$$\mathbf{M}_{error} = \frac{\|\Delta \mathbf{M}_{sys}\|}{\|\Delta \mathbf{M}_{sys,analytical}\|} = 2.483 \cdot 10^{-16} \sim 0 \quad (5.6)$$

The result in Eq. (5.6) reveals that the Gaussian quadrature provides a reliable approximation of Eq. (2.26).

5.1.3 Natural Frequencies

Following the validation of the $\mathbf{K}$ - and $\mathbf{M}$ -matrix, the remaining step is to derive the natural frequencies and compare them to an analytical solution. The natural frequencies of the free-free bar element, shown in Figure 5.1 are determined by solving the eigenvalue problem given by Eq. (3.74), employing the material properties from Table 5.1.

The application of the eigenvalue solver, `eigh` is demonstrated in Code 5.5. To optimize the efficiency of the process, it is advisable to restrict the solution to a subset of the matrix equation system, leveraging the sorting of known displacements, such as boundary conditions or supports. Furthermore, the function can optionally take a start and end index to return a chosen set of eigenvalues; e.g. `[0, 2]` to return the smallest 3 eigenvalues.



Code 5.5: Solve eigenvalue problem

```
# Solve the generalised eigenvalue problem if boundary conditions are
# specified
D, U = linalg.eigh(K[np.ix_(df,df)],M[np.ix_(df,df)], subset_by_index =
    solve_subset)

# Solve the generalised eigenvalue problem
D, U = linalg.eigh(K,M, subset_by_index = solve_subset)

# Convert Eig. Values to Natural Frequencies
omega = np.sqrt(D).real
```

Table 5.2: Relative errors of 2D longitudinal vibration for a free-free bar.

Mode	Code 5.5, 1 ele.	Code 5.5, 100 ele.
n = 1	0.1027	$4.112 \cdot 10^{-5}$
n = 2	0.1027	$1.645 \cdot 10^{-4}$
n = 3	0.1027	$3.702 \cdot 10^{-4}$

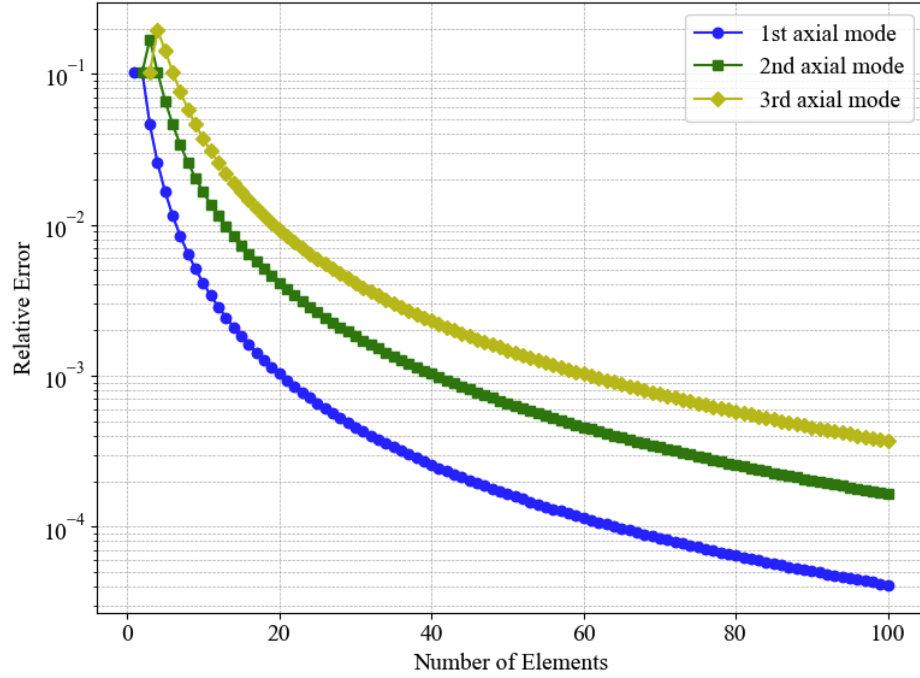


Figure 5.2: Relative frequency error against number of elements, 2D bar element.

In Inman (2007), analytical solutions for vibration frequencies in the longitudinal direction for various bar configurations are provided. The solution for the free-free bar in Figure 5.1 is given by

$$\omega_n = \frac{n\pi \sqrt{\frac{E}{\rho}}}{L} \quad (5.7)$$

Employing Eq. (5.7) and Code 5.5, the relative errors are obtained and presented in Table 5.2.

As demonstrated in Figure 5.2, the model exhibits convergence and approaches the analytical results, thereby validating the model's ability to approximate the analytical results. It is important to note that this convergence is only possible in this particular instance, due to $\mathbf{A} = \mathbf{I}$.

5.1.4 Transformed Bar Element in 2D Space

To validate the transformation matrix of the 2D bar element in Eq. (2.33), the element shown in Figure 5.1 is positioned in two-dimensional space using the nodal coordinates listed in Table 5.3.

Table 5.3: Nodal coordinates for 2D transformation test.

Node	x [m]	y [m]
1	0.000	0.000
2	19.22	5.000

Table 5.4: Validation of the 2D bar transformation matrix.

Configuration	1st Natural Frequency [rad/s]
Aligned (Reference, $\mathbf{A} = \mathbf{I}$)	867.0
Transformed Element	867.0

The validation is performed by comparing the first natural frequency of the transformed element with that of an aligned reference element, where the transformation matrix $\mathbf{A} = \mathbf{I}$. The two results in Table 5.4 are expected to be identical, thereby confirming the validity of the transformation procedure.

5.2 3D Truss

For the validation of 3D Truss the free-free bar element shown in Figure 5.1 is utilized.

5.2.1 The Stiffness Matrix

The implementation of the element stiffness matrix follows the same method as for 2D in Code 5.1, with the distinction of the updated $\mathbf{B}$ matrix. This results in a matrix, where the z -coordinate at each node is responsible for two additional columns and rows in the stiffness matrix. The numerical unified solution in Code 5.6 is determined using the properties detailed in Table 5.1.

The numerical stiffness matrix is evaluated by comparing it to the analytical formulation and calculating the error as in Eq. (5.3), resulting in

$$\mathbf{K}_{error} = 7.182 \cdot 10^{-11} \sim 0 \quad (5.8)$$

Indicating that the numerical and analytical stiffness matrices are well aligned.

5.2.2 The Mass matrix

As with the mass matrix in the 2D Truss program, an isometric implementation of the displacement interpolation matrix is used. An analytical solution of the mass matrix is presented in Eq. (2.26). To allow a direct comparison between Code 5.3 and the analytical solution, the system parameters from Table 5.1 are used. The norm in Eq. (5.6) is utilized to denote the magnitude of the disparity between the numerical solution in Code 5.6 and analytical matrix in Eq. (2.26)

$$\mathbf{M}_{error} = 7.182 \cdot 10^{-16} \sim 0 \quad (5.9)$$

The comparison indicates a high degree of consistency between the numerical and analytical results.



Code 5.6: 3D Truss - Unified longitudinal stiffness matrix

```
# K_uni = K * L / (E*A)
K_uni = [[ 1. -1.]
         [-1.  1.]]
```



Code 5.7: 3D Truss - Unified longitudinal mass matrix

```
# M_uni = M * 6 / (rho*A*L)
M_uni = [[2. 1.]
         [1. 2.]]
```

Table 5.5: Relative errors of 3D longitudinal vibration for a free-free bar.

Mode	Code 5.5, 1 ele.	Code 5.5, 100 ele.
n = 1	0.1027	$4.112 \cdot 10^{-5}$
n = 2	0.1027	$1.645 \cdot 10^{-4}$
n = 3	0.1027	$3.702 \cdot 10^{-4}$

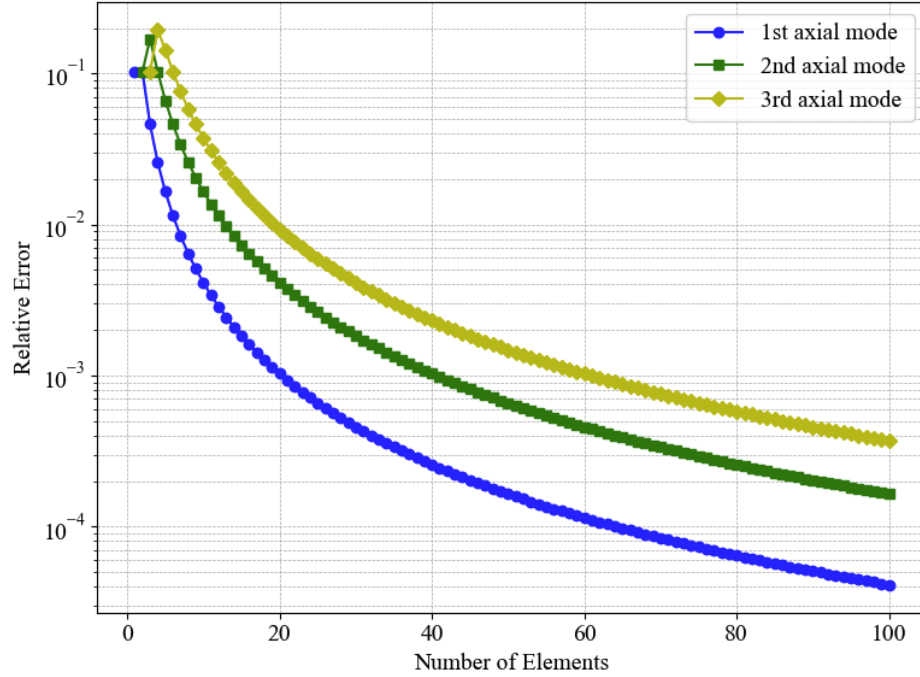


Figure 5.3: Relative frequency error against number of elements, 3D bar element.

5.2.3 Natural Frequencies

Ultimately the natural frequencies will be validated by numerically solving Eq. (3.74) using `scipy.linalg.eigh` in Code 5.5. The analytical solution from Inman (2007) addresses the eigenvalue problem in 1D. However, the natural frequencies of should not change from 1D to 3D if identical properties are used. Therefore, Eq. (5.7) is applied, utilizing the material properties specified in Table 5.1. Using Eq. (5.7) and Code 5.5, the relative errors are obtained and presented in Table 5.5.

As illustrated in Figure 5.3, the model demonstrates clear convergence towards the analytical solution, confirming its accuracy and reliability in approximating the expected results. As anticipated, the findings obtained from the 3D model are identical to those acquired from the 2D model.

5.2.4 Transformed Truss Element in 3D Space

To verify the transformation matrix in Eq. (2.52) for the 3D Truss program, the element shown in Figure 5.1 is used. The element is placed arbitrarily in three-dimensional space, with the nodal coordinates specified in Table 5.6.

Table 5.6: Nodal coordinates for 3D transformation test.

Node	x [m]	y [m]	z [m]
1	0.000	0.000	0.000
2	1.000	5.000	19.19

Table 5.7: Validation of the 3D bar transformation matrix.

Configuration	1st Natural Frequency [rad/s]
Aligned (Reference, $\mathbf{A} = \mathbf{I}$)	867.0
Transformed Element	867.0

The transformation is validated by comparing the first natural frequency to that of a reference element aligned along the global x -axis. Identical results in Table 5.7 confirm the validity of the implemented transformation.

5.3 2D Frame

For the validation of the 2D Frame model, a clamped-free cantilever beam is utilized. Only the transverse deformations and vibrations are considered in this section, as the axial behavior has already been validated previously. Since the axial formulation and implementation are identical, repeating the analysis would yield the same results.

5.3.1 The Stiffness Matrix

The element stiffness matrix is derived from the finite element formulation in Section 3.2. The iso-parametric definitions of the strain interpolation functions and the Gaussian quadrature method are employed. With respect to Eq. (3.32), the following modifications are made

$$\mathbf{k}_e = \int_0^L \mathbf{B}^T \mathbf{D} \mathbf{B} dx = |J| \int_{-1}^1 \mathbf{B}_{iso}^T \mathbf{D} \mathbf{B}_{iso} ds \quad (5.10)$$

The implementation of the equation is demonstrated in Code 5.8 and the analytical local element stiffness matrix for transverse displacements is shown in Eq. (3.32) and used for validation. In order to compare the analytical solution and the Python implementation, the parameters of the system are defined in Table 5.8. The beam is assumed to have a hollow circular cross-section, with geometric and material properties as defined for analysis in Table 6.1 and Table 6.2.

The element stiffness matrix for the beam in Figure 5.4 is directly obtained in global coordinates as $\mathbf{A} = \mathbf{I}$. The result in Code 5.9 is obtained using Code 5.8 followed by a subsequent filtration, that exclusively retains the transverse components. Using Eq. (3.32), the analytical solution is compared to the numerical solution in Code 5.9, resulting in

$$\mathbf{K}_{error} = 5.856 \cdot 10^{-11} \sim 0 \quad (5.11)$$

The calculation of the error validates the 2D Frame model, given that the two solutions are close to equivalent.

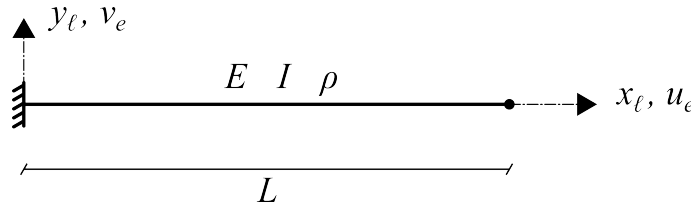


Figure 5.4: Clamped-free beam element.

Table 5.8: Material properties - clamped-free beam.

Element	A	I	E	L	ρ
Level A - Leg	$556.3 \cdot 10^{-3} \text{ m}^2$	$432.8 \cdot 10^{-3} \text{ m}^4$	210.0 GPa	19.86 m	8500 kg/m ³

**Code 5.8: 2D Frame - Stiffness matrix implementation**

```

A, L = Abeam(n1, n2)
xip, wip = intpL(PolDeg)
J = L/2          # Jacobi Determinant
D = np.array([[Ge[0]*Ge[1], 0],
              [0, Ge[0]*Ge[2]]])
# Define local stiffness matrix
k_l = np.zeros((6,6))
for i in range(len(xip)):
    B = Bint(xip[i],L)
    k_l += B.T @ D @ B * wip[i] * J

```

**Code 5.9: 2D Frame - Unified transversal stiffness matrix**

```

# K = K * L**3/(E*I)
K = [[ 12.          119.15999998 -12.          119.15999998]
      [ 119.15999998 1577.67839973 -119.15999998  788.83919987]
      [ -12.          -119.15999998  12.          -119.15999998]
      [ 119.15999998  788.83919987 -119.15999998 1577.67839973]]

```

5.3.2 The Mass Matrix

The mass matrix is implemented from the derivation of the finite element formulation in Section 3.2. The implementation method of the mass matrix is the same to that in Code 5.3. An analytical solution is provided in Eq. (3.33).

The numerical solution derived from the 2D Frame model is presented in Code 5.10, followed by a comparison with the analytical solution.

The computed error is given by

$$\mathbf{M}_{error} = 1.650 \cdot 10^{-10} \sim 0 \quad (5.12)$$

The numerical implementation thereby provides a reliable approximation of Eq. (3.33).

5.3.3 Natural Frequencies

Subsequent to validating the **K**- and **M**-matrix, the final step is to derive the natural frequencies and compare them with an analytical solution. The approach employed in this section is similar to that used in Section 5.1.3. Inman (2007) provides analytical solutions to frequencies in the transverse direction for various beam configurations. In this analysis, the primary focus will be on the first three natural frequencies in the transverse direction of the clamped-free beam in Figure 5.4.

**Code 5.10: 2D Frame - Unified transversal mass matrix**

```

# M = M * 420/(rho*A*L)
M = [[ 156.00000001  436.92000002  54.          -258.18000001]
      [ 436.92000002 1577.67840009 258.18000001 -1183.25880007]
      [ 54.          258.18000001 156.00000001 -436.92000002]
      [-258.18000001 -1183.25880007 -436.92000002 1577.67840009]]

```

Table 5.9: Relative errors of 2D transverse vibration for a clamped-free beam.

Mode	Code 5.5, 1 ele.	Code 5.5, 100 ele.
n = 1	$4.754 \cdot 10^{-3}$	$9.773 \cdot 10^{-7}$
n = 2	$8.486 \cdot 10^{-3}$	$2.909 \cdot 10^{-9}$
n = 3	$12.46 \cdot 10^{-3}$	$2.896 \cdot 10^{-8}$

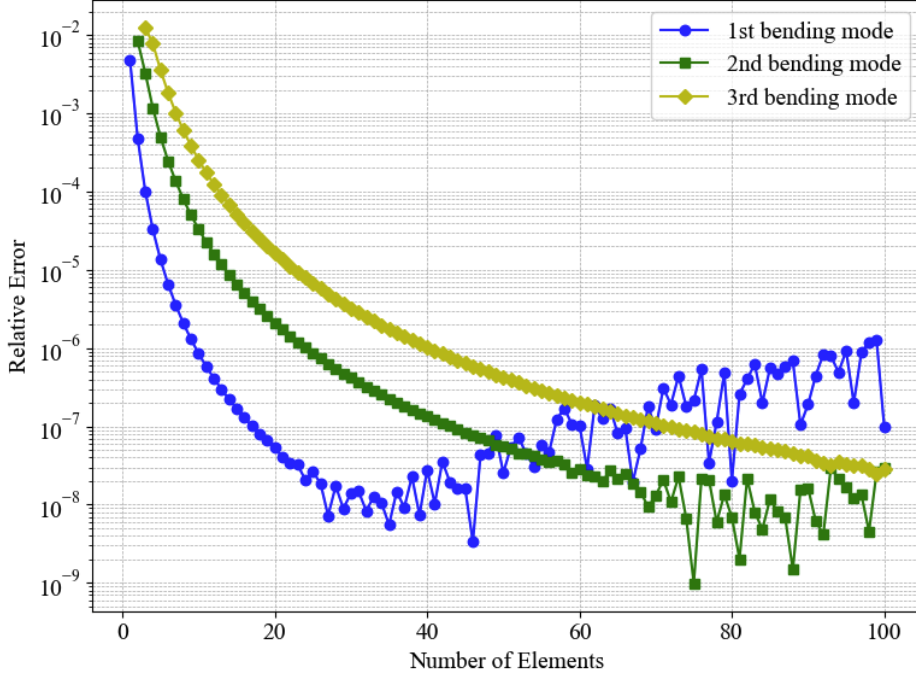


Figure 5.5: Relative frequency error against number of elements, 2D beam element.

The solution using weighted natural frequencies is given by

$$\omega_n = \beta_n^2 \sqrt{\frac{EI}{\rho A}} \quad (5.13)$$

The material parameters used are the same as specified in Table 5.8. Using Eq. (5.13) and Code 5.5, the relative error in Table 5.9 is obtained.

As demonstrated in Figure 5.5, the model exhibits convergence and approaches the analytical results, thereby validating the model's ability to approximate the analytical results. The first bending mode stops converging at 24 elements, with an relative error of approximately $2 \cdot 10^{-8}$. The other modes stop converging around the same value.

5.3.4 Transformed Frame Element in 2D Space

To validate the transformation matrix in Eq. (3.35) for the 2D Frame element, the configuration illustrated in Figure 5.4 is employed. The element is positioned arbitrarily in two-dimensional space using the nodal coordinates listed in Table 5.3. Consistent results in Table 5.10 confirm the validity of the transformation procedure.

Table 5.10: Validation of the 2D beam transformation matrix.

Configuration	1st Natural Frequency [rad/s]
Aligned (Reference, $\mathbf{A} = \mathbf{I}$)	39.27
Transformed Element	39.27

Table 5.11: Material properties - clamped-free beam (torsion).

Element	J	G
Level A - Leg	$865.7 \cdot 10^{-3} \text{ m}^4$	81.00 GPa

5.4 3D Frame

The 2D beam element in Figure 5.4 undergoes a transformation to the 3D-space, resulting in an element with 12 DOFs. The beam is assumed to have properties as defined in Table 5.8. Furthermore, given that the model is three-dimensional, the shear modulus and torsional constant are defined in Table 5.11.

5.4.1 The Stiffness Matrix

The analytical 3D beam element stiffness matrix is given in Eq. (3.66). The development of the numerical 3D beam stiffness matrix is analogous to 2D, demonstrated in Code 5.8, with the distinction of modifying the dimensions of $\mathbf{k}_e$.

Axial and transverse vibration are confirmed to remain consistent with those previously obtained, as a result of the identical implementation in the program. Therefore, the focus is solely on the torsional vibration. The torsional stiffness matrix for a beam element is derived analytically in Eq. (3.54).

To compare the analytical solution with the Python implementation, the system parameters from Table 5.8 and Table 5.11 are employed. The torsional stiffness matrix in Code 5.11 is obtained using an extended version of Code 5.8.

The error for the stiffness matrix is

$$\mathbf{K}_{error} = 7.182 \cdot 10^{-11} \sim 0 \quad (5.14)$$

The result shows agreement between the numerical and analytical mass matrices, confirming the validity of the implementation.

5.4.2 The Mass Matrix

As previously stated, the element has 12 degrees of freedom, resulting in a 12x12 mass matrix. However, for this comparison, the matrix is reduced to include only the torsional elements.

For the example illustrated in Figure 5.4, an analytical solution for the mass matrix is available in Eq. (3.55). The torsional mass matrix is computed using an extended version of Code 5.3.



Code 5.11: 3D Frame - Unified torsional stiffness matrix

```
# K_uni = K_torsion * L / (G * J)
K_uni = [[ 1. -1.]
         [-1.  1.]]
```



Code 5.12: 3D Frame - Unified torsional mass matrix

```
# M_uni = M_torsion * 6/(rho*J*L)
M_uni = [[2. 1.]
         [1. 2.]]
```

The error between the result in Code 5.12 and the mass matrix in Eq. (3.55) is

$$\mathbf{M}_{error} = 7.182 \cdot 10^{-11} \sim 0 \quad (5.15)$$

The result obtained demonstrates high consistency between the numerical and analytical mass matrices, confirming the validity of the implementation.

5.4.3 Natural Frequencies

The methodology applied for validating the natural frequencies mirrors the approach described in Section 5.1.3. Inman (2007) presents the analytical solution for the clamped-free beam element for torsional beam vibrations

$$\omega_n = \frac{(2n - 1)\pi\sqrt{\frac{G}{\rho}}}{2L} \quad (5.16)$$

which is used to evaluate the natural frequencies associated with Eq. (3.74). This analysis focuses on the first two torsional modes of the clamped-free 3D beam configuration shown in Figure 5.4. Applying Eq. (5.16) and comparing to the Python implementation yields the relative errors in Table 5.12.

The convergence of the relative error is analyzed numerically in Python and shown in Figure 5.6. As with Figure 5.5, the first bending mode stops converging at an relative error of approximately $2 \cdot 10^{-8}$.

5.4.4 Transformed Beam Element in 3D Space

To validate the transformation matrix for the 3D beam element presented in Eq. (3.68), the element is arbitrarily positioned in the three-dimensional space with the nodal coordinates defined in Table 5.6. The transformation matrix implementation is confirmed to be accurate, as evidenced by the identical results in Table 5.13.

Table 5.12: Relative errors of 3D torsional vibration for a clamped-free beam.

Mode	Code 5.5, 1 ele.	Code 5.5, 100 ele.
n = 1	$1.027 \cdot 10^{-1}$	$1.025 \cdot 10^{-5}$
n = 2	1.42310^{-1}	9.25310^{-5}

Table 5.13: Validation of the 3D beam transformation matrix.

Configuration	1st Torsional Frequency [rad/s]
Aligned (Reference, $\mathbf{A} = \mathbf{I}$)	268.8
Transformed Element	268.8

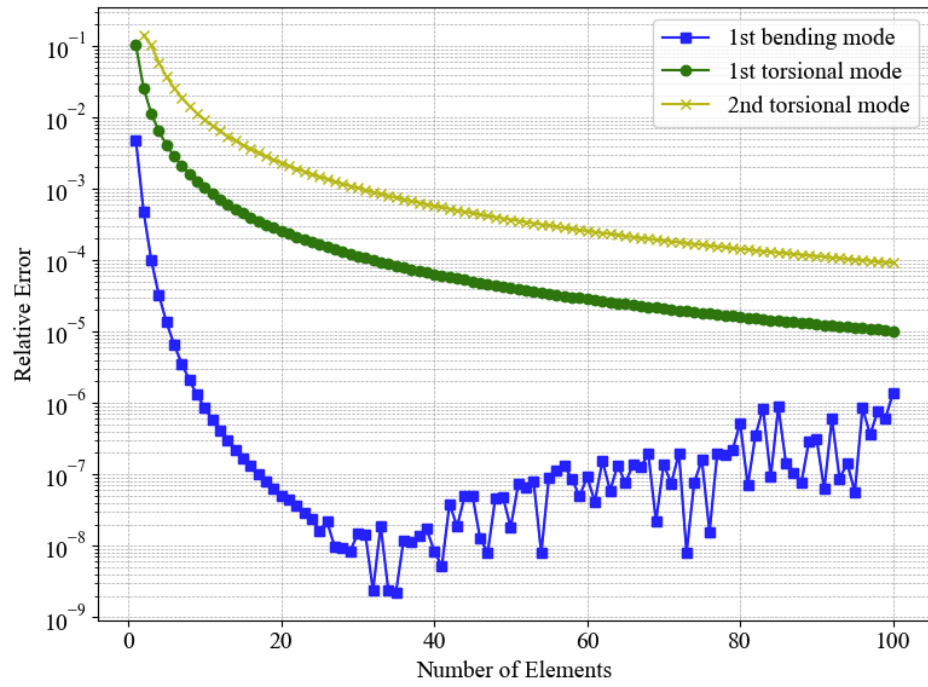


Figure 5.6: Relative frequency error against number of elements, 3D beam element.

6 Computational Analysis of Jacket Models

Following the validation of the programs for a single element, the analysis is extended to the jacket structure. Subsequently, the scope of the study is expanded to an analysis of the combined system formed by jacket, tower and RNA. The computation time of each program is evaluated considering models with and without the tower structure. Finally, the convergence of the frame programs is investigated.

6.1 Natural Frequencies of the Jacket Structure

This section presents and analyzes the natural frequencies and associated mode shapes of the jacket structure across the four different modeling approaches. The design utilized during the analysis and modes are categorized into global and local structural modes, to distinguish overall structural behavior from member-level dynamics. However, the analysis will focus solely on the global modes, as they govern the behavior of the global system.

6.1.1 Design of the Jacket Structure

The jacket foundation design is based on the support structure of the 20MW wind turbine, as described in the report by Pontow et al. (2017). For the initial analysis of natural frequencies and mode shapes both the jacket and transition piece is considered. This is done in order to best assimilate the frequencies of the substructure. Instead of piles, the jacket is simply supported at the base to prevent displacement in two or three directions. The 2D projection of the jacket structure is visualized in Figure 6.1.

The jacket foundation is a four-legged lattice structure. Each of the four sides is supported by four levels of X-braces, referred to as Levels A–D.

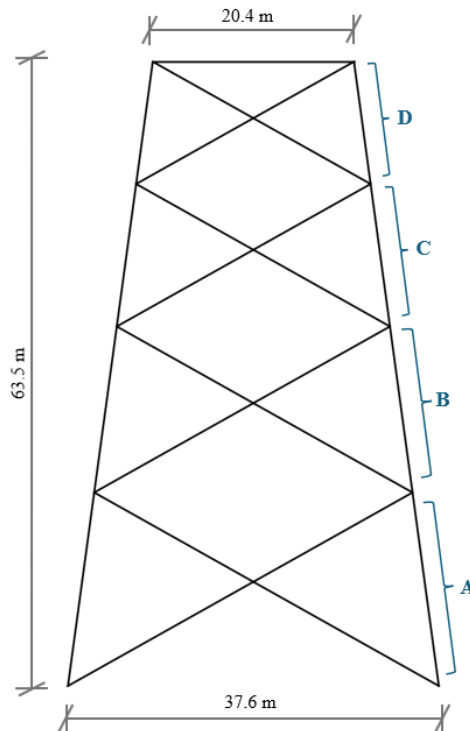


Figure 6.1: Main geometric properties of the 20MW reference jacket.

Table 6.1: Geometric properties for leg and brace elements.

Level - Element	Outer diameter [mm]	Wall thickness [mm]
A - Brace	1168	32
B - Brace	965	22
C - Brace	965	25
D - Brace	914	20
A - Leg	2438	51
B - Leg	1828	45
C - Leg	1829	45
D - Leg	1829	51

The main geometric properties are the height of the jacket, the width at mudline and the width at the top of the jacket given by Figure 6.1. These properties are found from the node coordinates in Pontow et al. (2017), also given in Appendix B.1. The 2D representation is a projection of one of the four sides from the 3D structure, resulting in equivalent heights. It is assumed that each of the four X-braces has an equal angle with the leg. Consequently, the length of the leg and brace elements decreases through level A-D.

The elements are hollow rods, defined by an outer diameter and a wall thickness, given by the preliminary design of Pontow et al. (2017) and in Appendix B.2. Joints are not considered for the geometry in the input data. The geometric properties are given in Table 6.1.

Cross-section area A , second moment of area I , and torsional constant J are determined from the geometric properties, when constructing the material dictionary. The material properties given in Table 6.2 apply to all steel elements in the jacket structure. The model's density parameters are in alignment with Pontow et al. (2017), while other steel properties such as the Young's modulus E , shear modulus G and Poisson's ratio ν are taken from Danish Standard Association (2022). Since the 2D structure represents only one side of the jacket, it should be noted that its mass is reduced compared to the 3D structure.

Transition piece

The transition piece is implemented in each program by hard-coding main properties in the structural framework, as described by Section 4.3. In Pontow et al. (2017) the transition piece height and total mass are specified, while an assumed width of 1.0 m is chosen in order to distribute the weight of the transition piece, see Table 6.3. The second moment of area, torsional constant, and distributed density are predefined in the programs.

In both 2D and 3D, the transition piece's entire mass is considered. In the 2D models, the transition piece is represented as a single element between the two top nodes, whereas in 3D models,

Table 6.2: Material parameters for steel.

	Symbol	Value
Density	ρ	8500 kg/m ³
Elastic Modulus	E	210 GPa
Shear Modulus	G	87.8 GPa
Poisson's ratio	ν	0.3

Table 6.3: Transition piece properties.

	Symbol	Value
Height transition piece	h_p	8 m
Width transition piece	b_p	1 m

Table 6.4: Mass of models.

	2D Models	3D Models	ROSAP
Jacket	503 ton	1368 ton	1670 ton
Transition piece	430 ton	430 ton	450 ton

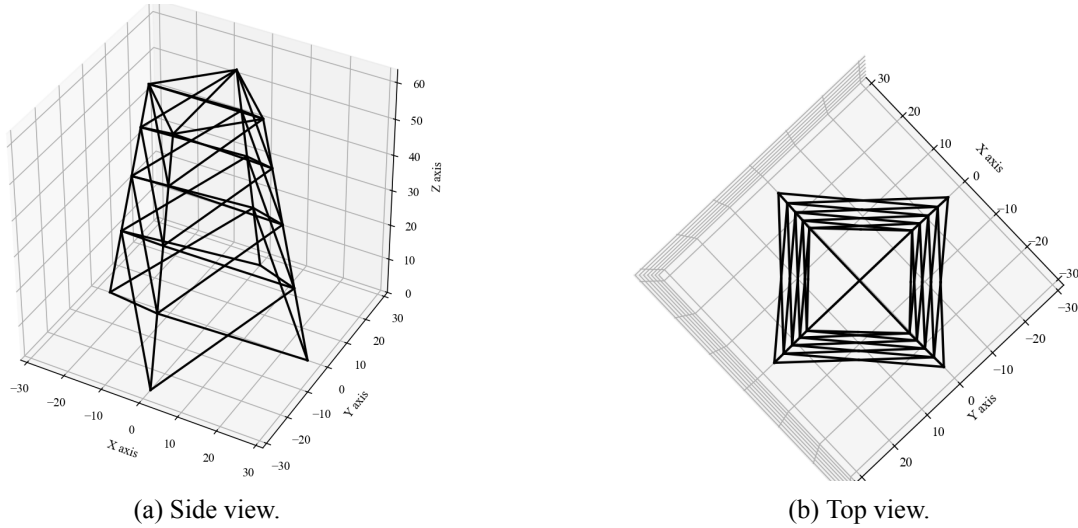


Figure 6.2: 3D Jacket and transition piece.

all four top nodes are linked along the sides and diagonals, see Figure 6.2 and Table 6.4. This implementation ensures mass consistency with Pontow et al. (2017) and an infinitely stiff transition piece.

6.1.2 Global Structural Modes

The analysis of the FE programs focuses on identifying global modes, as they primarily govern the overall structural behavior. Special attention is given to the lowest natural frequencies, which are most critical for dynamic responses to environmental loads such as wind and waves. In the subsequent chapters, the term 'jacket' will be used to describe the combined jacket-transition piece structure.

Global Bending Modes

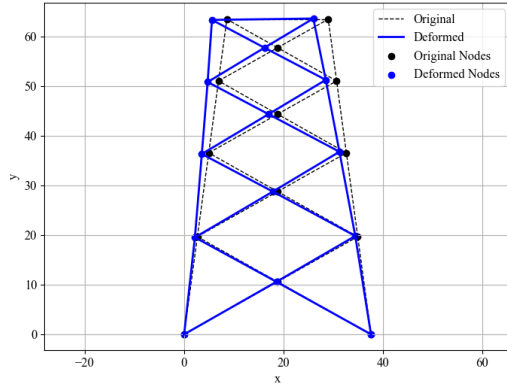
The global bending modes represent the dominant lateral deformation patterns of the entire jacket structure. Table 6.5 lists the first four global bending modes across all models. The first bending mode occurs in the range of approximately 2.9 Hz to 3.5 Hz.

The plots for the first bending mode from each of the programs are presented in Figure 6.3 and Figure 6.4. All mode shapes used in this analysis are available in the Digital Appendix D.2. In the 3D models, bending modes appear in pairs, corresponding to bending about the x - and y -axes. Due to the symmetry of the jacket structure, these mode pairs have identical natural frequencies.

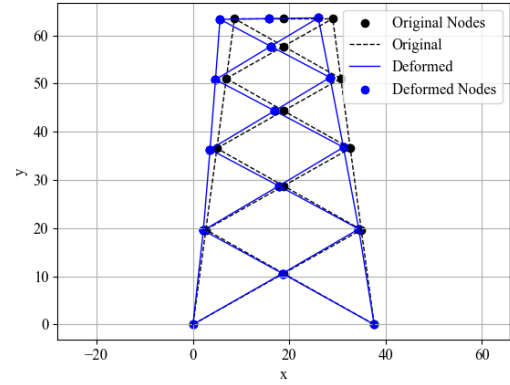
The fourth global bending mode appears as the 64th mode in the 3D Frame model. This high mode number is due to the presence of numerous local member modes, particularly from brace elements. Further subdividing frame elements introduces additional local modes, pushing global bending modes even higher in the mode sequence.

Table 6.5: Global bending mode frequencies [Hz] for jacket with corresponding system mode numbers.

Bending Mode	2D Truss		2D Frame		3D Truss		3D Frame	
1	2.900	1	2.933	1	3.484	1	3.496	4
2	10.54	3	10.32	3	9.694	5	10.26	22
3	20.88	7	16.65	6	17.11	13	16.35	44
4	35.14	9	20.71	10	19.01	20	20.40	64

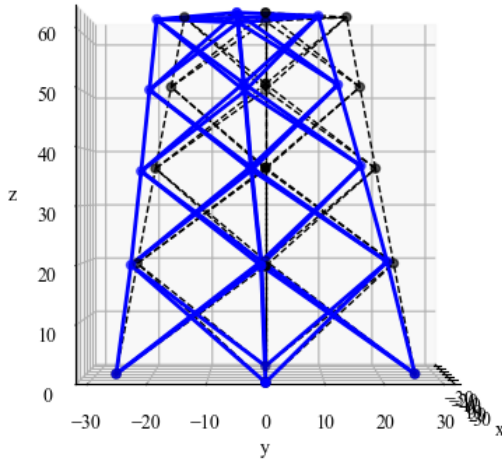


(a) 2D Truss, mode shape 1 - scaled by 3.

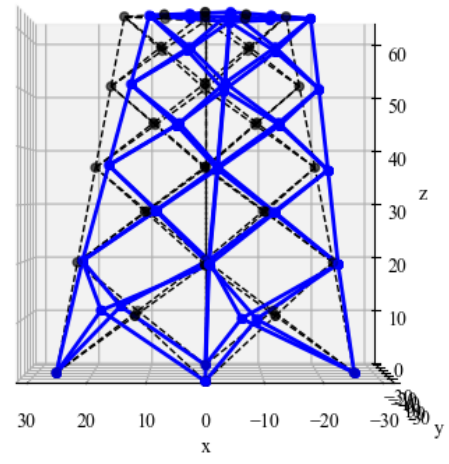


(b) 2D Frame, mode shape 1 - scaled by 3.

Figure 6.3: 1st bending mode - 2D models.



(a) 3D Truss, mode shape 1 - scaled by 5.



(b) 3D Frame, mode shape 4 - scaled by 6.

Figure 6.4: 1st bending mode - 3D models.

The 3D Frame program serves as a benchmark, as it is the most complex model accounting for axial, transverse and torsional effects. As shown in Table 6.5, the first bending mode occurs at mode 4, with a natural frequency of 3.496 Hz. The 3D Truss model demonstrates a close alignment, with a relative discrepancy of 0.34%. The first bending mode in the 2D programs appears at lower frequencies, with relative differences of 17% and 16% for the truss and frame models, respectively.

For the second bending mode, the 2D models show better agreement with the 3D Frame model than the 3D Truss does. In bending modes 3-4, the 2D Frame model provides the closest approximation to the 3D Frame results, with a maximum relative difference of 1.8%.

Global Axial Modes

Global axial modes, involving vertical displacements along the longitudinal axis of the jacket, are presented in Table 6.6. A comparison of the frequencies for the first global axial mode reveals that the 2D models exhibit a reduced natural frequency in comparison to the 3D models. The relative difference from the 3D Frame model is 13% and 14%, respectively, for the 2D Truss and 2D Frame model. The 3D Truss provides a close approximation of the 3D Frame's natural frequency, yielding only a relative difference of 0.096%. However, the axial mode is not considered critical, as it is unlikely to be excited by naturally occurring dynamic loads.

Table 6.6: Global axial mode frequencies [Hz] for jacket with corresponding system mode numbers.

Axial Mode	2D Truss	2D Frame	3D Truss	3D Frame
1	9.067 2	9.032 2	10.45 9	10.46 24

Table 6.7: Global torsional mode frequencies [Hz] for jacket with corresponding system mode numbers.

Torsional Mode	3D Truss	3D Frame
1	5.342 3	5.359 12
2	12.44 12	9.541 21

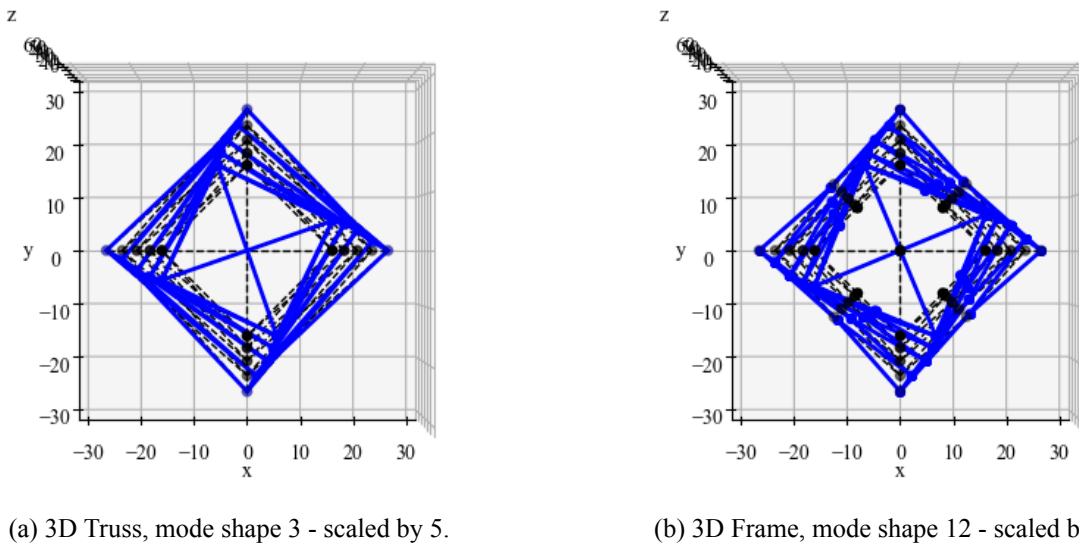


Figure 6.5: 1st torsional mode.

Global Torsional Modes

The first torsional mode appears at approximately 5.35 Hz, with a slight difference of 0.32% transitioning from the truss to the frame model, as presented in Figure 6.5 and Table 6.7. The second torsional mode is significantly higher in the 3D Truss model than in the 3D Frame model, with a relative difference of 30%. When more modes are considered, it is observed that the 3D Frame model captures more torsional modes compared to the 3D Truss.

6.2 Natural Frequencies of the Jacket-Wind Turbine System

Additionally, to capture the dynamic response of the combined system interaction, the jacket is analyzed together with a tower and a simplified representation of the Rotor-Nacelle-Assembly (RNA).

6.2.1 Design of Tower and RNA

The tower is based on the 20 MW reference design according to Pontow et al. (2017), and divided into nine tapered sections, as summarized in Table 6.8 (see also Appendix B.3). Standard steel properties are used for the tower as outlined in Table 6.2.

The RNA is modeled as a concentrated mass and torsional inertia applied to the tower's top node, with properties defined in Table 6.9. The RNA mass is provided in Pontow et al. (2017). In

Table 6.8: Tower geometry.

Element	Outer diameter [mm]	Wall thickness [mm]	Length [m]	Height [m]
1	10950	51	6.14	6.14
2	10556	48	16.26	22.4
3	10163	45	16.27	38.7
4	9769	42	16.27	54.9
5	9375	40	16.26	71.2
6	8981	37	16.26	87.5
7	8587	34	26.26	103.7
8	8193	31	16.26	120.0
9	7777	28	17.17	137.1

Table 6.9: Tower and RNA properties.

	Symbol	Value
Tower mass	m_t	1356 ton
Rotor mass moment of inertia	J_r	$2919.66 \cdot 10^6 \text{ kg m}^2$
Rotor mass	m_r	1730 ton

addition, the rotor mass moment of inertia is given by the article by Gambier and Meng (2019).

An initial NFA is conducted with the tower modeled as a cantilever beam, which is done to enable comparison of the tower's global modes when it is placed on the substructure. Results are presented in Table 6.10. Bending and axial modes are well aligned between the two models. Torsional modes only appear in the 3D model.

6.2.2 Global Structural Modes

The global modes of the combined system are analyzed. Modes are categorized by whether they are primarily introduced by the tower (T) or the jacket (J). In some cases, the response is coupled, with both the tower and jacket contributing significantly to the mode shape.

Global Bending Modes

As shown in Table 6.11, the lowest two bending modes are dominated by tower motion, closely matching the tower results in Table 6.10, indicating that the stiffness of the jacket has a limited effect on the first tower modes. The third and fourth bending modes are associated with a coupled response. Similar to the jacket-only case, the 2D models yield lower natural frequencies for the first bending mode compared to the 3D Frame model. However, unlike the results in Table 6.5, the 2D models here consistently underestimate the natural frequencies relative to the 3D Frame. Among all programs, the 3D Truss provides the closest approximation to the 3D Frame results. The

Table 6.10: Global tower mode frequencies [Hz].

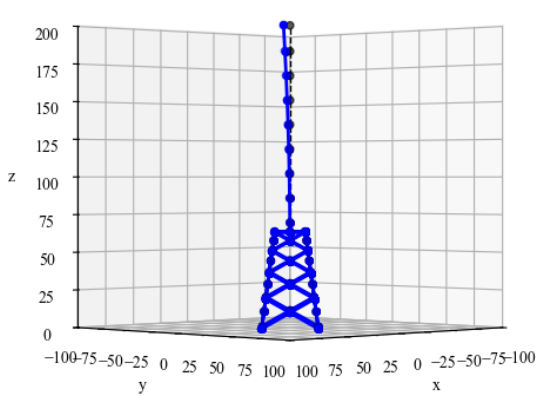
Mode Type	2D Frame		3D Frame	
Bending 1	0.2040	1	0.2040	1
Torsion 1	-	-	0.3217	3
Bending 2	0.6607	2	0.6607	4
Bending 3	3.223	3	3.223	6
Axial 1	4.523	4	4.523	8

Table 6.11: Global bending mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.

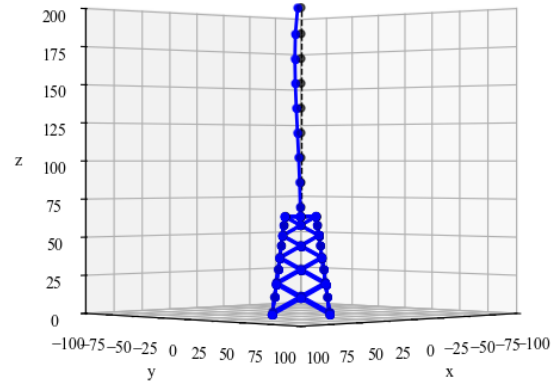
Bending Mode	Intro. by	2D Truss		2D Frame		3D Truss		3D Frame	
1	T	0.1815	1	0.1806	1	0.1915	1	0.1903	1
2	T	0.5955	2	0.5944	2	0.6218	4	0.6196	4
3	J/T	1.844	3	1.858	3	2.213	6	2.216	6
4	J/T	3.579	5	3.586	5	3.829	9	3.773	13

corresponding bending mode shapes from 3D Frame are illustrated in Figure 6.6 and Figure 6.7.

A common trend across all three models is that the largest relative difference from the 3D Frame occurs at the third global bending mode, which results from a coupled interaction between the tower and jacket. The relative differences are as follows: 17% for the 2D Truss, 16% for the 2D Frame, and 1.5% for the 3D Truss.

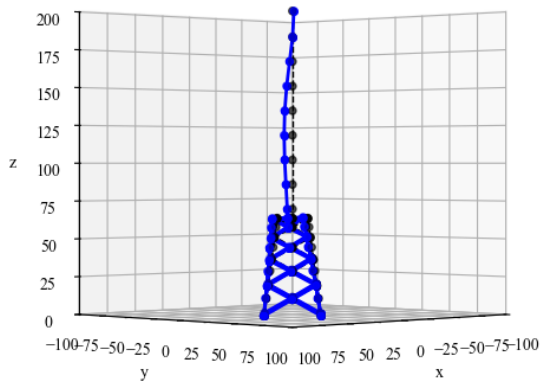


(a) 3D Frame, mode shape 1 - scaled by 6.

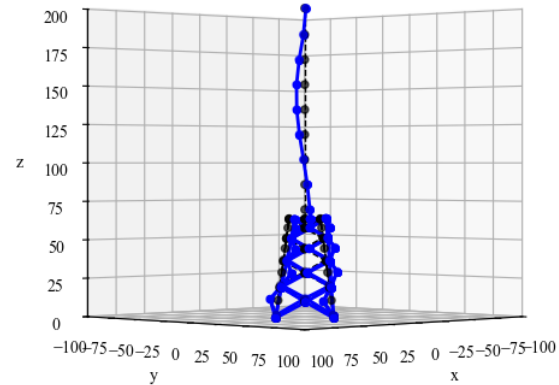


(b) 3D Frame, mode shape 4 - scaled by 6.

Figure 6.6: 1st and 2nd global bending mode - combined jacket, tower & RNA - 3D Frame.



(a) 3D Frame, mode shape 6 - scaled by 6.



(b) 3D Frame, mode shape 13 - scaled by 6.

Figure 6.7: 3rd and 4th global bending mode - combined jacket, tower & RNA - 3D Frame.

Table 6.12: Global axial mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.

Axial Mode	Intro. by	2D Truss	2D Frame	3D Truss	3D Frame
1	T	2.947 4	2.919 4	3.499 8	3.466 11
2	J	11.49 7	9.253 7	9.876 15	9.867 33

Table 6.13: Global torsional mode frequencies [Hz] for combined jacket-wind turbine with corresponding system mode numbers.

Torsional Mode	Intro. by	3D Truss	3D Frame
1	T	0.3078 3	0.2972 3
2	J	5.143 11	5.148 20

Global Axial Modes

Table 6.12 shows the global axial modes of the combined system. A comparison with Table 6.10 and Table 6.6 reveals the origin of the axial modes in the combined system. The first axial mode in Table 6.12 closely matches the first axial mode of the tower in Table 6.10, while the second axial mode aligns well with the first axial mode of the jacket structure in Table 6.6.

Axial modes are generally less affected by the dimensionality of the model, and their frequencies remain relatively stable across different configurations. However, a notable discrepancy is observed between the 2D Truss and 3D Frame model for the second axial mode. As shown in Table 6.12, the 2D Truss predicts a significantly higher frequency in mode 2 compared to the 3D Frame, resulting in the largest relative difference of 16% among all models.

Global Torsional Modes

The torsional behavior of the complete system is only captured in the 3D models, shown in Table 6.13. Compared to the jacket, the inclusion of the tower significantly lowers the first torsional frequency. However, if the second torsional mode is compared to the first torsional mode of the jacket structure, the frequency is only reduced with approximately 0.2 Hz.

6.3 Computation Time

The program efficiency is evaluated based on computation time. For each model the time required, to assemble the system matrices and solve the first 20 eigenvalues, is recorded and averaged over 10,000 computations. When considering the tower, the entire matrix system must be solved for the truss models, as no subset can be applied. These operations are executed within the VIBtruss/VIBframe class. The resulting computation times are presented in Table 6.14 and Table 6.15.

Table 6.14: Average computation time over 10,000 runs and relative speed for the jacket structure, using the 2D Truss model as the benchmark. Computations (1) and (2) were performed on separate hardware.

	2D Truss	2D Frame	3D Truss	3D Frame
(1) Computation time [10^{-3} s]	2.244	4.062	11.32	46.23
(1) Relative speed	1	1.810	5.047	20.61
(2) Computation time [10^{-3} s]	1.801	3.057	8.349	38.90
(2) Relative speed	1	1.697	4.637	21.60

Table 6.15: Average computation time over 10,000 runs and relative speed for the combined jacket, tower, and RNA system, using the 2D Truss jacket model as the benchmark. Computations (1) and (2) were performed on separate hardware.

	2D Truss	2D Frame	3D Truss	3D Frame
(1) Computation time [10^{-3} s]	4.611	6.078	19.85	57.01
(1) Relative speed	2.055	2.709	8.849	25.41
(2) Computation time [10^{-3} s]	3.518	5.293	15.58	53.51
(2) Relative speed	1.954	2.939	8.651	29.72

The computation time is recorded on two computers and then compared as a relative speed with jacket model in 2D Truss as the benchmark. This is due to the fact that the computation is dependent on the hardware. The same benchmark is employed for the coupled system, thus facilitating a comparison of the computation time across models.

As demonstrated in Table 6.14 and Table 6.15, the computational time increases across the four programs, from 2D Truss to 3D Frame. The tables also show that the incorporation of the tower and RNA has the most significant impact on the computation time of the truss models, particularly the 2D Truss, which approximately doubles its computation time. As expected, the computation times differ between the two test runs performed on separate hardware. However, the relative speed provides a basis for comparing performance across models, independent of the absolute computation time.

6.4 Convergence of the Frame Models

Convergence is evaluated for both frame models by examining their first two global modes, using up to 32 element subdivisions. In the 2D Frame model, these are the first bending and first axial mode, corresponding to system mode 1 and 2. In the 3D Frame model, the first global modes are the first bending and first torsional mode, corresponding to mode 4 and 12.

The convergence results are presented in Table 6.16 and Table 6.17. Both tests solve for modes 1 through 12 to ensure comparable computation times. The computation time is averaged over multiple run, which is particularly important for models with less element meshing, as they have shorter computation times.

The convergence is shown as the relative difference from the reference value at 32 element subdivisions, plotted against the number of element subdivisions. The results in Figure 6.8 show clear convergence for element mesh refinements of 1, 2, 4, 8, and 16 subdivisions. The linear trend in the double-log plot indicates a power-law relationship, confirming convergence behavior. This linearity is most pronounced in the initial refinements.

Table 6.16: Convergence results for 2D Frame.

No. el.	No. runs	Comp. time [s]	1st Bending Mode	1st Axial Mode
1	10,000	0.004130	2.93311561	9.03158965
2	10,000	0.009599	2.93259601	9.00491926
4	10,000	0.02490	2.93255993	9.00285176
8	1000	0.06870	2.93255696	9.00265669
16	1000	0.23144	2.93255659	9.00262832
32	100	0.9607	2.93255665	9.00262249

Table 6.17: Convergence results for 3D Frame.

No. el.	No. runs	Comp. time [s]	1st Bending Mode	1st Torsional Mode
1	10,000	0.05439	3.49579246	5.35887366
2	10,000	0.1435	3.48194320	5.34184735
4	1000	0.5647	3.48102605	5.34070015
8	100	3.270	3.48096682	5.34061633
16	10	24.05	3.48096277	5.34060822
32	1	178.6	3.48096250	5.34060719

The computation time versus the number of element subdivisions is shown in the log-log plot in Figure 6.9. The 2D Frame requires significantly less computation time than the 3D Frame, primarily due to the lower number of DOFs. For reference, the computation time for 8 elements in the 2D model is comparable to that of a single element in the 3D model.

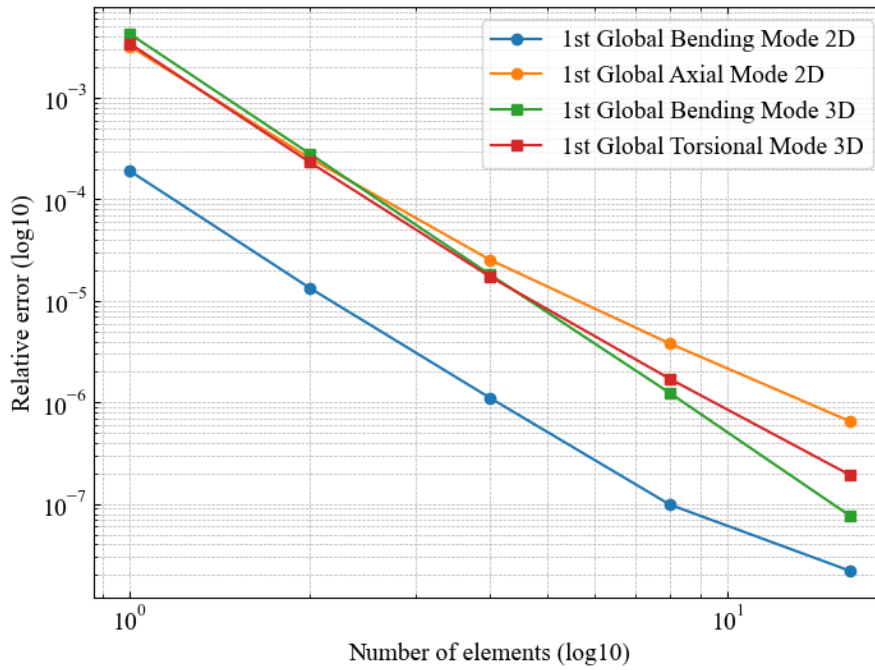


Figure 6.8: Convergence of relative difference to value at 32 elements against number of subdivisions of elements.

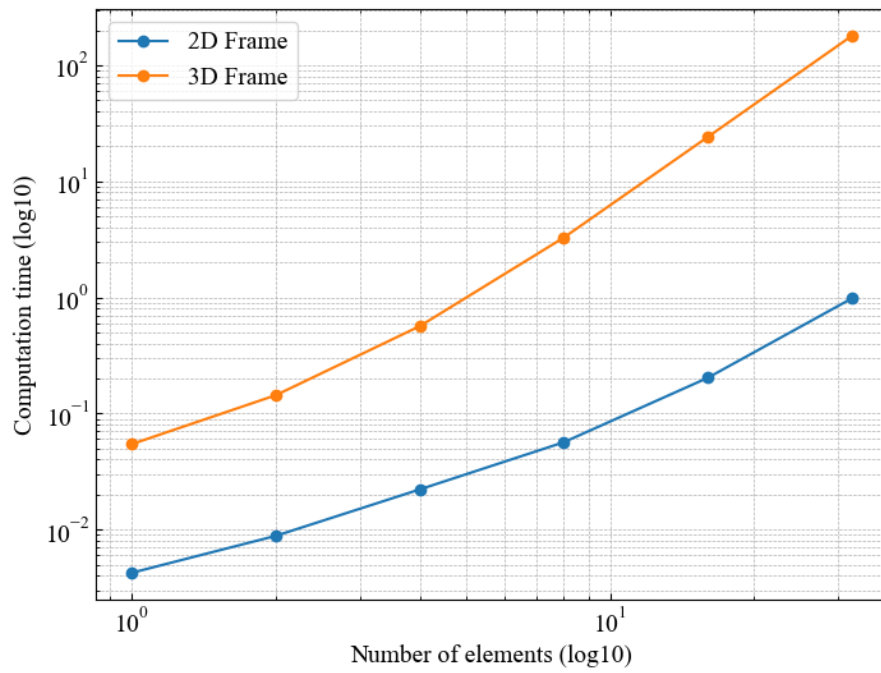


Figure 6.9: Computation time against number of element subdivisions.

7 Evaluation of the Computational Models

The computational analysis is evaluated, firstly focusing on the jacket structure and subsequently on the combined system incorporating the tower and RNA. The evaluation includes an analysis of natural frequencies and computation time for each model. In addition, the convergence behavior of the frame programs is assessed to determine stability and reliability across different mesh resolutions.

7.1 Program Assessment for Jacket Frequency Analysis

The 3D Frame program will serve as the benchmark in the evaluation of the modeled jacket structure, as it is the most complex program incorporating axial, transverse, and torsional effects.

7.1.1 Evaluation of 2D Models' Performance Against 3D Frame

A comparison of the 2D Truss model to 3D Frame shows that the first global mode, for both the axial and bending mode, present a reduced natural frequency. The relative difference is 17% and 13% for the axial and bending mode, respectively. Additionally, for the subsequent modes 2 through 4 the 2D Truss under perform in accurately capturing the correct natural frequencies when comparing to the 3D model.

The 2D Frame model present the same trend for the first global axial and bending mode. The natural frequencies of the first global axial and bending modes are reduced by 16% and 14%, respectively. The 2D Frame provides the best approximation of the 3D Frame's bending modes 2 through 4 across all three programs, with a maximum difference of 1.8%.

The inaccuracy of the 2D models may be explained by several factors. A fundamental limitation of the 2D structure is that it is only able to represent a single side of a four-sided jacket, which is projected into a 2D plane. Consequently, the height of the models remains consistent between the 2D and 3D models, but the length of the elements is reduced slightly, leading to a slightly increased stiffness.

Furthermore, the jacket's total mass has undergone a substantial reduction, decreasing from 1368 tons in the 3D model to 503 tons in the 2D models, accounting for only 37% of the full mass. Despite the transition piece mass being equivalent in both models, it constitutes a much larger portion of the total mass. Consequently, the utilization ratio of the jacket legs in the 2D models is significantly higher than that of the 3D models. This disproportion leads to an imbalance between mass and stiffness, which alters the system's dynamic response.

Another source of inaccuracy in the 2D models is their inability to capture out-of-plane behavior. In Figure 6.4 the 3D models first bending mode primarily involves displacement about a single axis. However, a minor secondary contribution from bending about the perpendicular axis is also observed, concurrent with local member vibrations. This coupled behavior cannot be represented in 2D, which limits the accuracy of the dynamic response. Despite this, the 2D Frame model appears to reasonably approximate the 3D Frame results beyond the first global bending mode, as demonstrated in Table 6.5. In contrast, the 2D Truss model shows larger deviations and is less accurate in capturing the correct natural frequencies.

7.1.2 Evaluation of 3D Truss Performance Against 3D Frame

The bending frequency results exhibit varying degrees of alignment between the 3D Truss and 3D Frame models. While the first global bending mode differs by only 0.34%, indicating strong correspondence, modes 2 through 4 show moderate variation, with frequencies fluctuating slightly around those of the 3D Frame. These deviations are likely due to the influence of local member

interactions present in the frame model, which the truss model cannot replicate. A limiting factor is the absence of mid-brace node connections, which restricts the 3D Truss model from capturing local brace behavior. Additionally, this results in some mode types being more difficult to categorize, as the 3D Truss model is less flexible.

In terms of torsional behavior, the 3D Truss model captures the first global torsional mode with great accuracy, showing a relative difference of just 0.32% compared to the 3D Frame. However, the second torsional mode differs significantly, with a 30% deviation. This suggests that frame elements provide a more accurate representation of complex torsional modes.

Finally, the first global axial mode shows great agreement, with a relative difference of 0.34%. This indicates that the 3D Truss model provides a reliable approximation of the first axial mode for full 3D Frame structure.

7.2 Program Assessment for Jacket-Turbine Frequency Analysis

The evaluation of the combined structure, comprising of the jacket, tower and RNA, will include a comparison to the natural frequency analysis presented in Pontow et al. (2017). Furthermore, the results of the combined structure across all three programs will be evaluated with the 3D Frame as benchmark.

7.2.1 Natural Frequency Results - ROSAP

Pontow et al. (2017) presents a natural frequency analysis of the 20 MW reference wind turbine and support structure, assessed using Ramboll's Offshore Structural Analysis Program (ROSAP). The analysis employs a structural model of the support structure, which integrates a simplified Rotor-Nacelle-Assembly, represented by point masses. The ROSAP implementation accounts for a number of factors which can influence the stiffness of the structure, including corrosion, marine growth, local scour and water level. Furthermore, the ROSAP model incorporates pile foundations and soil stiffness. The model provides a more comprehensive dynamic analysis, which will serve as a benchmark for comparing the results of the developed programs.

In Pontow et al. (2017) three distinct tests are conducted for three different stiffness configurations. The test used as a benchmark will be the stiffest configuration, which does not include corrosion, marine growth, nor local scour. The model does account for water level at lowest astronomical tide (LAT). Given that the model employed in this study is derived from the referenced report, a comparative analysis of the determined natural frequencies is feasible. The natural frequencies obtained from Pontow et al. (2017) are summarized in Table 7.1, and the corresponding modes shape to the global modes are illustrated in Figure 7.1.

7.2.2 Evaluation of 3D Frame

A comparative analysis is conducted between the global bending modes from ROSAP and the 3D Frame model. The natural frequencies and the corresponding mode shape numbers are presented in Table 7.2. The first and second global bending modes are primarily governed by the tower structure. When modeled as a clamped cantilever beam in the Frame program, the tower yields natural frequencies of 0.2040 Hz and 0.6607 Hz for the first two bending modes. In the 3D Frame model, the first coupled bending mode appears at 0.1903 Hz, representing a decrease of approximately

Table 7.1: Natural frequency for first 10 modes. Pontow, S.,Kaufer, D., Shirzadeh, R. & Kühn M. (2017), *Design Solution for a Support Structure Concept for future 20MW*, INNWIND.EU, p. 34

4x4-LSTIF	M.1 (Hz)	M.2 (Hz)	M.3 (Hz)	M.4 (Hz)	M.5 (Hz)	M.6 (Hz)	M.7 (Hz)	M.8 (Hz)	M.9 (Hz)	M.10 (Hz)
	0.1632	0.1640	0.8254	0.9303	1.0005	1.5978	1.6352	2.4639	2.4673	2.4796

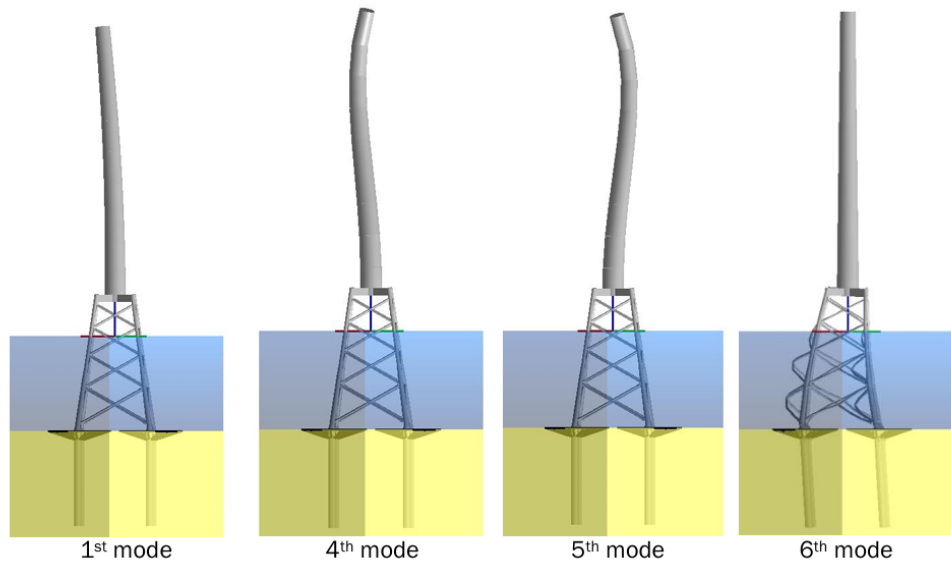


Figure 7.1: First eight eigenmodes of the jacket structure. Pontow, S.,Kaufer, D., Shirzadeh, R. & Kühn M. (2017), *Design Solution for a Support Structure Concept for future 20MW*, IN-NWIND.EU, p. 34

Table 7.2: Comparison of frequencies to global bending modes in ROSAP and 3D Frame, with corresponding mode number.

	1st Bending (T)	2nd Bending (T)	3rd Bending (J/T)	4th Bending (J/T)
ROSAP	0.1632 Hz 1	0.903 Hz 4	1.0005 Hz 5	1.6352 Hz 6
3D Frame	0.1903 Hz 1	0.6196 Hz 4	2.216 Hz 6	3.773 Hz 13

0.01 Hz. By comparison, the ROSAP model predicts the first bending mode at 0.1632 Hz, which may suggest that the jacket structure in the 3D Frame model provides greater overall stiffness. One possible explanation for this difference lies in the boundary conditions, as the ROSAP model incorporates soil stiffness, while the 3D Frame model assumes a simple supported base. Additionally, the discrepancy may be partly due to differences in the tower modeling, as the applied J_{RNA} is kept unknown in Pontow et al. (2017).

The ROSAP model generally exhibits significantly lower natural frequencies, suggesting reduced stiffness and/or higher mass compared to the 3D Frame model, as demonstrated in Table 7.2. However, the same conclusion cannot be drawn for the second bending mode, as the ROSAP model predicts a frequency approximately 0.3 Hz higher than that of the 3D Frame. Notably, this second bending mode of the ROSAP model also exceeds the second bending mode of the tower in Table 6.10. This finding suggests that the mill model used in ROSAP differs from that employed in the 3D Frame, resulting in a stiffer effective system within this mode range.

The 3D Frame model provides a reasonable representation of the first and second bending mode shapes when compared to the ROSAP model. In contrast, the third and fourth bending modes show clear differences between the two models. The mode shapes are not consistently aligned, and the natural frequencies begin to diverge significantly. This growing deviation suggests that the models differ more substantially in how they capture the dynamic behavior of the structure at higher modes, likely due to differences in how the jacket and tower are modeled.

In summary, these results indicate that the 3D Frame model is suitable for capturing the fundamental dynamic characteristics of the system but becomes less reliable in reproducing higher-order bending behavior. Further investigation into model assumptions and input parameters may help reduce these discrepancies.

7.2.3 Evaluation of Programs Performance Against 3D Frame

A comparison between the 2D models and the 3D Frame model for the combined jacket-tower system reveals that the 2D models consistently predict lower natural frequencies across all four global bending modes, as shown in Table 6.11. A contributing factor may be the fixed tower and RNA mass, which constitutes a larger proportion of the total system mass in the 2D models. This shifts the overall mass–stiffness balance, resulting in lower natural frequency estimates.

However, in contrast to the jacket-only case, the 2D models show improved agreement with the 3D Frame results, particularly for the first and second bending modes. This improvement is likely because these modes are primarily governed by the tower, which is modeled using beam elements in all four programs. As a result, the influence of jacket simplifications in the 2D models is reduced, leading to more accurate frequency estimates.

The third global bending mode shows the largest deviation from the 3D Frame model among the 2D models, with relative differences of 17% and 16% for the 2D Truss and 2D Frame, respectively. The third mode is primarily influenced by the jacket structure, making it more sensitive to the simplifications inherent in the 2D representations.

Across all programs, the 3D Truss model provides the closest overall alignment with the 3D Frame, particularly for the first two global bending and torsional modes. The computed natural frequencies show only minor deviations, with a maximum relative error of 1.5% on global bending mode 3. This highlights that, despite its simplified element formulation, the truss based model remains effective in approximating the global modes of the full frame system with the tower and RNA included.

7.3 Model Benchmarking and Computation Time

As part of the results in Section 6.3 the computational efficiency of the programs has been evaluated based on computation time. The results in Table 6.14 and 6.15 indicate a clear trend; computation time increases with the model complexity from the simplest model to the most complex.

7.3.1 FLOPs and Computation Time

An effective measure of algorithmic efficiency is counting floating point operations (FLOPs). It is important to highlight that FLOPs are not directly measured here, but would have served a better overview of the models complexity and computation efficiency, as FLOPs are hardware independent.

It has not been possible to track FLOPs on any of the programs due to the implementation, which has led to tracking computation times. The offered FLOPs counting tool for Python is not supported by Windows and require a Linux based system (Flozz, 2025). While absolute computation times vary between computation (1) and (2) due to differences in processor speed, the relative speeds remain fairly consistent, showcased in Table 6.14. This suggests that while computation time is hardware-dependent, relative performance between model types is a more stable and comparable metric. However, even when averaged over 10,000 runs, there is still some inherent uncertainty.

7.3.2 Computational Efficiency and Utility Evaluation

As discussed in Section 7.1, the 3D Truss model provides the best approximation of the 3D Frame results. It also offers a significant computational advantage, reducing computation time to 25% of the 3D Frame when analyzing the jacket structure.

While the 2D Frame model offers a good approximation of higher-order modes beyond the first global bending mode, its practical utility is limited, as the lowest natural modes are typically the most critical. However, it does offer a significant computational advantage, running at roughly 10% the speed of the 3D Frame model.

As shown in Section 7.2, the addition of the tower and RNA improves frequency approximation across all models. However, it also leads to a significant increase in computation time. This effect is more pronounced in the truss models, where the number of DOFs increases significantly due to the use of beam elements for the tower structure, combined with a more computationally demanding eigenvalue solver. Despite this increase, the 3D Truss program still completes analysis in approximately 30–35% of the time required by the 3D Frame model.

7.4 Convergence of the Frame Models

Both the 2D Frame and 3D Frame program show convergence, as visualized in Figure 6.8. The evaluation focuses on the jacket structure only, since the tower implementation does not support element subdivision in the current implementation.

The 3D Frame model is limited to a maximum of 64 element subdivisions due to matrix size constraints, which restrict solver capacity. However, additional refinement beyond this point provides minimal benefit. In the program validation and Figure 5.5 it is shown that the solutions stops converging when the error becomes less than $2 \cdot 10^{-8}$.

Interestingly, the first bending mode of the 3D Frame model, once converged, aligns more closely with the result from the 3D Truss program in Table 6.5. The same trend is observed in Table 6.7 for the first torsional mode. While the available data is limited, this pattern suggests that the 3D Truss model may serve as a reliable approximation of the fully converged 3D Frame behavior, particularly for lower modes. Furthermore, this observation strengthens the argument for using the 3D Truss program to approximate the 3D Frame.

8 Conclusion

The primary objective of this project was to develop an in-house finite element program for determining the natural frequencies and mode shapes of jacket foundations. The development aimed at enhancing usability and reducing application time through parametric user inputs for the jacket design. To investigate whether simpler models could serve as alternatives to a full-scale 3D Frame model, four model types were implemented: 2D Truss, 2D Frame, 3D Truss, and 3D Frame. Each model was assessed for its accuracy in predicting natural frequencies and mode shapes, as well as its computational efficiency.

The 3D Frame program was evaluated by comparing its results against a reference jacket from Pontow et al. (2017). While its natural frequencies showed noticeable deviations, resulting from stiffness and mass discrepancies in the jacket and tower, the 3D Frame model still accurately reproduced the first two mode shapes. This suggests that refinements to the input data is needed for improved frequency accuracy.

The comparative analysis of the 2D Truss, 2D Frame and 3D Truss against the 3D Frame highlights key differences between accuracy and computational efficiency. The 3D Truss model is closely matching the 3D Frame's results for the first global bending, axial, and torsional frequencies, with relative errors below 0.35%. Its simplified formulation, using only bar elements, reduces computation time to roughly 25% of the 3D Frame, making it an efficient choice for early-stage dynamic assessments.

The 2D Frame model shows better alignment with the 3D Frame results for global bending modes 2 through 4. However, it underestimates the first global bending frequency by 16%, cannot capture torsional behavior, and suffers from geometric simplifications that lead to significant mass discrepancies. The 2D Truss model, while the fastest, exhibits larger deviations and the lowest accuracy overall, making it the least suitable option for dynamic analysis.

The utility of the programs improves significantly with the inclusion of the tower and RNA. Modeling the tower using beam elements and applying Lagrange's method of undetermined multipliers in the truss models results in closer agreement with the 3D Frame model in terms of both tower response and overall system dynamics. The 3D Truss program shows a relative difference of 1.5% or less compared to the 3D Frame model across all evaluated global modes. Despite the added degrees of freedom introduced by the tower and the use of Lagrange multipliers, the 3D Truss model achieves a computation time of approximately 30–35% of the 3D Frame model.

The analysis of the developed models shows that a simplified 3D Truss model can closely approximate natural frequencies of a 3D Frame model for a combined wind turbine and support structure. This study demonstrates that using simplified models with bar elements can significantly reduce computation time while maintaining reliable approximations. These findings support the possibility of using simpler modeling approaches for preliminary design and experimental testing.

9 Future Work

While this project successfully developed and analyzed computational models for structural dynamics, several aspects remain open for further exploration and refinement. This final chapter outlines and discusses potential scopes for future work, focusing on enhancing model accuracy and expanding the applications of the developed programs.

Model and Experimental Testing

The estimation of natural frequencies differs significantly from the benchmark established by the ROSAP analysis in the report by Pontow et al. (2017) and further testing of the model is required. The parametric implementation facilitates analysis through systematic variation of parameters, assessing its effect on natural frequencies. It should be examined whether the present implementation can accurately approximate experimentally determined natural frequencies, or if additional model updates are required.

Improve Model Accuracy

A key issue in the 2D models is the mass discrepancy when transitioning from 3D to 2D. In order to improve model accuracy, the models should be tested across a range of mass up- and down-scaling in 2D, and performing a comparative analysis to the 3D Frame model.

The 3D Truss program for the jacket–transition piece system exhibits notable discrepancies compared to the 3D Frame model in higher-order global bending modes. A limitation lies in the absence of node connections at the intersections of X-braces in the truss model, which prevents the capture of local brace-related modes. To address this, virtual restraints can be introduced to reestablish connectivity, either through spring elements or by employing Lagrange multipliers, as successfully implemented in the tower modeling. However, the additional computational cost introduced by this modification should be assessed to determine whether the accuracy gains justify the increased complexity.

Additionally, the input data could be further refined by applying incremental adjustments to the geometric properties of individual elements. Such enhancements may help improve frequency predictions, particularly in higher modes where sensitivity to geometric stiffness becomes more pronounced.

Extended Applications

Integrating a parametric implementation for the tower is a logical next step in extending the model. This enhancement would enable user input and adaptation, allowing analysis using parametric variations of the tower structure.

Additionally, a considered extension of the program could include the option of applying loads to the structure, specifically gravity and temperature loading. Gravity loading would allow the optimization of the design by analyzing stress in the elements at the bottom levels. Furthermore, temperature loading would support the objectives of the Hybrid Wind project, with the final goal of determining the influence of cold climate effects.

With the application of loads, the models could further be extended to incorporate a solver for computing section forces, enabling identification of critical elements in the jacket design. Additionally, it would allow the implementation of a geometric stiffness matrix to the truss model, aiming to more accurately approximating the behavior of a frame model.

Bibliography

- Cook, R., Malkus, D., & Pleashe, M.** (1989). *Concepts and applications of finite element analysis* (3rd ed.). Wiley.
- Danish Standard Association.** (2022). Eurocode 3 - Design of steel structures - part 1-1: General rules and rules for buildings [EN 1993-1-1].
- DTU.** (2025). Hybridwind - DTU [Accessed: 07 June 2025]. <https://hybridwind.dtu.dk/>
- European Commission.** (2025). *In focus: Wind energy powering the clean transition* [Accessed: 2025-04-10]. https://energy.ec.europa.eu/news/focus-wind-energy-powering-clean-transition-2025-03-17_en
- Flozz.** (2025). Pypapi installation guide [Accessed: 12 June 2025]. <https://flozz.github.io/pypapi/install.html>
- Gambier, A., & Meng, F.** (2019). Control system design for a 20MW reference wind turbine. *Proceedings of the 2019 IEEE Conference on Control Technology and Applications (CCTA)*, 258–263. <https://doi.org/10.1109/CCTA.2019.8920600>
- Inman, D.** (2007). *Engineering vibration* (3rd). Pearson Education.
- Krenk, S.** (2018). *Dynamics of structures*. Technical University of Denmark.
- Krenk, S., & Høgsberg, J.** (2012). *Statics and mechanics of structures*. Springer.
- Logan, D.** (2021). *A first course in the finite element method* (6th). Cengage.
- Nielsen, J., Olesen, J., & Poulsen, P.** (2019). *Bærende konstruktioners numeriske modellering - elementmetoden* (1st). DTU Byg.
- NumPy Developers.** (2025). numpy.linalg.norm — Numpy v2.2 manual [Accessed: 2025-04-14]. <https://numpy.org/doc/2.2/reference/generated/numpy.linalg.norm.html>
- Pontow, S., Kaufer, D., Shirzahdeh, R., & Kühn, M.** (2017). *Deliverable D4.36: Design solution for a support structure concept 20MW* (tech. rep.). INNWIND.EU.
- Ramboll.** (2025). Green growth and sustainable living [Accessed: 24 April 2025]. <https://history.ramboll.com/cases/green-growth-and-sustainable-living>
- Santelo, T., de Oliveira, C., Maciel, C. D., & Monteiro, J.** (2022). Wind turbine failures review and trends. *Journal of Control, Automation and Electrical Systems*, 33, 505–521. <https://doi.org/10.1007/s40313-021-00789-8>
- The SciPy Community.** (2025). scipy.linalg.eigh — SciPy v1.15.2 manual [Accessed: 2025-04-06]. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.eigh.html>

A Lagrange Multipliers

A.1 Coupling of Jacket Foundation and Tower Model in 2D

The theoretical foundation for the method of Lagrange's multipliers is outlined in **Cook et al. (1989)**. This appendix demonstrates the theoretical formulation used to couple the 2D Truss model representing the jacket foundation with a 2D Frame model representing the wind turbine tower structure. The two substructures are modeled independently and coupled through the method of Lagrange multipliers.

A.1.1 Equations of Motion

The governing equation of motion for the tower structure is written in matrix form as

$$\bar{\mathbf{M}}\ddot{\mathbf{r}} + \bar{\mathbf{K}}\mathbf{r} = 0 \quad (\text{A.1})$$

and

- $u_1 = \mathbf{d}_1^T \mathbf{r}$,
- $u_2 = \mathbf{d}_2^T \mathbf{r}$,
- $u_3 = \mathbf{d}_3^T \mathbf{r}$.

where $\mathbf{d}_i$ is a zero vector with $\mathbf{d}[i] = 1$.

The equation of motion for the jacket structure is written in matrix form as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = 0 \quad (\text{A.2})$$

and

- $q_1^x = (\mathbf{b}_1^x)^T \mathbf{q}$,
- $q_1^y = (\mathbf{b}_1^y)^T \mathbf{q}$,

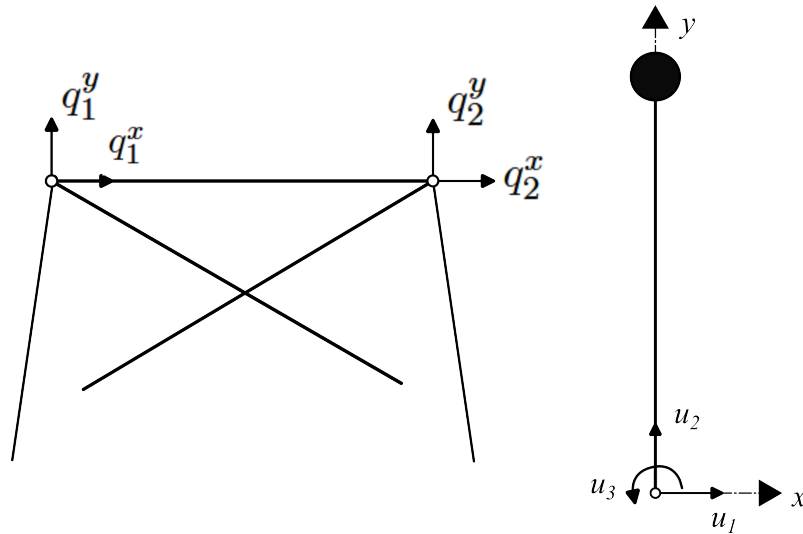


Figure A.1: DOF coupled through Lagrange multipliers. (Left: Jacket-Structure, Right: Tower-Structure)

- $q_2^x = (\mathbf{b}_2^x)^T \mathbf{q}$,
- $q_2^y = (\mathbf{b}_2^y)^T \mathbf{q}$.

where $\mathbf{b}_i$ is a zero vector with $\mathbf{b}[i] = 1$.

A.1.2 Coupling Conditions

The coupling conditions are defined such that the displacements of the tower's base and the top nodes of the jacket are equal in the x and y directions, and rotational compatibility is ensured through an angular constraint. These constraints are expressed as

$$u_1 = \frac{1}{2} (q_1^x + q_2^x) = \frac{1}{2} ((\mathbf{b}_1^x)^T + (\mathbf{b}_2^x)^T) \mathbf{q} \quad (\text{A.3})$$

$$\Rightarrow \begin{bmatrix} \frac{1}{2} ((\mathbf{b}_1^x)^T + (\mathbf{b}_2^x)^T) & -\mathbf{d}_1^T \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_1^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.4})$$

$$u_2 = \frac{1}{2} (q_1^y + q_2^y) = \frac{1}{2} ((\mathbf{b}_1^y)^T + (\mathbf{b}_2^y)^T) \mathbf{q} \quad (\text{A.5})$$

$$\Rightarrow \begin{bmatrix} \frac{1}{2} ((\mathbf{b}_1^y)^T + (\mathbf{b}_2^y)^T) & -\mathbf{d}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_2^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.6})$$

$$u_3 = \frac{1}{L} (q_2^y - q_1^y) = \frac{1}{L} ((\mathbf{b}_2^y)^T - (\mathbf{b}_1^y)^T) \mathbf{q} \quad (\text{A.7})$$

$$\Rightarrow \begin{bmatrix} \frac{1}{L} ((\mathbf{b}_2^y)^T - (\mathbf{b}_1^y)^T) & -\mathbf{d}_3^T \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_3^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.8})$$

Collecting all constraint equations into one matrix form yields

$$\mathbf{w} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3] \quad (\text{A.9})$$

The system matrices are augmented to incorporate the Lagrange multipliers

$$\mathbf{M}_\lambda = \begin{bmatrix} \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \bar{\mathbf{M}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{K}_\lambda = \begin{bmatrix} \begin{bmatrix} \mathbf{K} & 0 \\ 0 & \bar{\mathbf{K}} \end{bmatrix} & \mathbf{w} \\ \mathbf{w}^T & 0 \end{bmatrix} \quad (\text{A.10})$$

The constraints are included via the augmented system

$$(\mathbf{K}_\lambda - \omega^2 \mathbf{M}_\lambda) \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \\ \lambda \end{bmatrix} = \mathbf{0} \quad (\text{A.11})$$

The described formulation enables a modular approach to modeling wind turbine foundations by coupling finite element models via constraints.

A.2 Coupling of Jacket Foundation and Tower Model in 3D

This appendix extends the theoretical formulation used for the 2D case to a 3D setting, where both the jacket foundation and the tower structure are modeled in three dimensions. The coupling is again handled via the method of Lagrange multipliers, enforcing displacement and rotational continuity between the two systems.

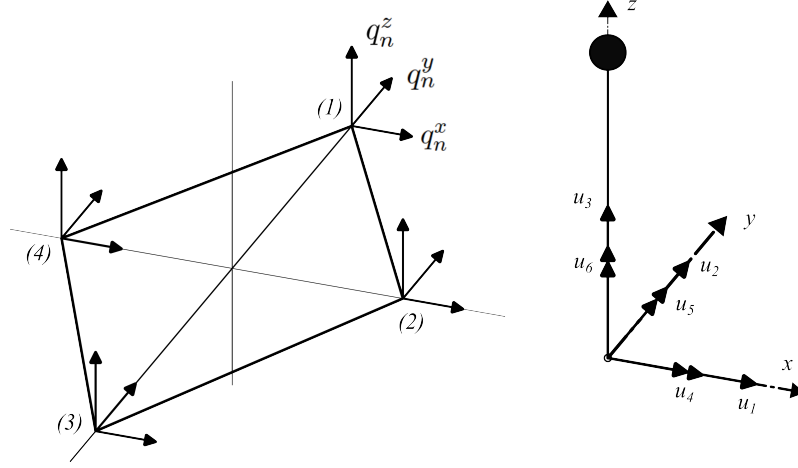


Figure A.2: DOF coupled through Lagrange multipliers. (Left: Jacket-Structure, Right: Tower-Structure)

A.2.1 Equations of Motion

The governing equation of motion for the tower structure is

$$\bar{\mathbf{M}}\ddot{\mathbf{r}} + \bar{\mathbf{K}}\mathbf{r} = 0 \quad (\text{A.12})$$

and

- $u_1 = \mathbf{d}_1^T \mathbf{r},$
- $u_2 = \mathbf{d}_2^T \mathbf{r},$
- $u_3 = \mathbf{d}_3^T \mathbf{r},$
- $u_4 = \mathbf{d}_4^T \mathbf{r},$
- $u_5 = \mathbf{d}_5^T \mathbf{r},$
- $u_6 = \mathbf{d}_6^T \mathbf{r}.$

The governing equation for the jacket foundation is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = 0 \quad (\text{A.13})$$

and

- $q_n^x = (\mathbf{b}_n^x)^T \mathbf{q},$
- $q_n^y = (\mathbf{b}_n^y)^T \mathbf{q},$
- $q_n^z = (\mathbf{b}_n^z)^T \mathbf{q},$

for $n = 1, 2, 3, 4.$

A.2.2 Coupling Conditions

The coupling is enforced at four top nodes of the jacket. Let these nodes be labeled (1), (2), (3), (4). The constraints are derived as

Translation in x -direction

$$u_1 = \frac{1}{4} (q_1^x + q_2^x + q_3^x + q_4^x) = \frac{1}{4} ((\mathbf{b}_1^x)^T + (\mathbf{b}_2^x)^T + (\mathbf{b}_3^x)^T + (\mathbf{b}_4^x)^T) \mathbf{q} \quad (\text{A.14})$$

$$\Rightarrow \left[\frac{1}{4} ((\mathbf{b}_1^x)^T + (\mathbf{b}_2^x)^T + (\mathbf{b}_3^x)^T + (\mathbf{b}_4^x)^T) \quad -\mathbf{d}_1^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_1^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.15})$$

Translation in y -direction

$$\left[\frac{1}{4} ((\mathbf{b}_1^y)^T + (\mathbf{b}_2^y)^T + (\mathbf{b}_3^y)^T + (\mathbf{b}_4^y)^T) \quad -\mathbf{d}_2^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_2^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.16})$$

Translation in z -direction

$$\left[\frac{1}{4} ((\mathbf{b}_1^z)^T + (\mathbf{b}_2^z)^T + (\mathbf{b}_3^z)^T + (\mathbf{b}_4^z)^T) \quad -\mathbf{d}_3^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_3^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.17})$$

Rotation about x -axis Only nodes (1) and (3) are used

$$u_4 = \frac{1}{L_{13}} (q_1^z - q_3^z) = \frac{1}{L_{13}} ((\mathbf{b}_1^z)^T - (\mathbf{b}_3^z)^T) \mathbf{q} \quad (\text{A.18})$$

$$\Rightarrow \left[\frac{1}{L_{13}} ((\mathbf{b}_1^z)^T - (\mathbf{b}_3^z)^T) \quad -\mathbf{d}_4^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_4^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.19})$$

Rotation about y -axis Only nodes (2) and (4) are used

$$u_5 = \frac{1}{L_{24}} (q_4^z - q_2^z) = \frac{1}{L_{24}} ((\mathbf{b}_4^z)^T - (\mathbf{b}_2^z)^T) \mathbf{q} \quad (\text{A.20})$$

$$\Rightarrow \left[\frac{1}{L_{24}} ((\mathbf{b}_4^z)^T - (\mathbf{b}_2^z)^T) \quad -\mathbf{d}_5^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_5^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.21})$$

Torsion about z -axis The constraint is formulated using average slopes between diagonally opposite nodes

$$\frac{1}{L_{13}} (q_3^x - q_1^x) = a_y, \quad \frac{1}{L_{24}} (q_2^y - q_4^y) = a_x \quad (\text{A.22})$$

These give a formulation for a sixth constraint u_6 , which can be generalized as

$$\Rightarrow \left[\frac{1}{2} \left(\frac{1}{L_{13}} ((\mathbf{b}_3^x)^T - (\mathbf{b}_1^x)^T) + \frac{1}{L_{24}} ((\mathbf{b}_2^y)^T - (\mathbf{b}_4^y)^T) \right) \quad -\mathbf{d}_6^T \right] \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \mathbf{w}_6^T \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \end{bmatrix} = 0 \quad (\text{A.23})$$

All constraints are collected in

$$\mathbf{w} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3 \quad \mathbf{w}_4 \quad \mathbf{w}_5 \quad \mathbf{w}_6] \quad (\text{A.24})$$

The augmented matrices are assembled as

$$\mathbf{M}_\lambda = \begin{bmatrix} \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \bar{\mathbf{M}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{K}_\lambda = \begin{bmatrix} \begin{bmatrix} \mathbf{K} & 0 \\ 0 & \bar{\mathbf{K}} \end{bmatrix} & \mathbf{w} \\ \mathbf{w}^T & 0 \end{bmatrix} \quad (\text{A.25})$$

Finally, the coupled system becomes

$$(\mathbf{K}_\lambda - \omega^2 \mathbf{M}_\lambda) \begin{bmatrix} \mathbf{q} \\ \mathbf{r} \\ \lambda \end{bmatrix} = \mathbf{0} \quad (\text{A.26})$$

This fully coupled system ensures compatibility in all 6 DOFs across the interface between the jacket and tower model in 3D.

B Jacket Design

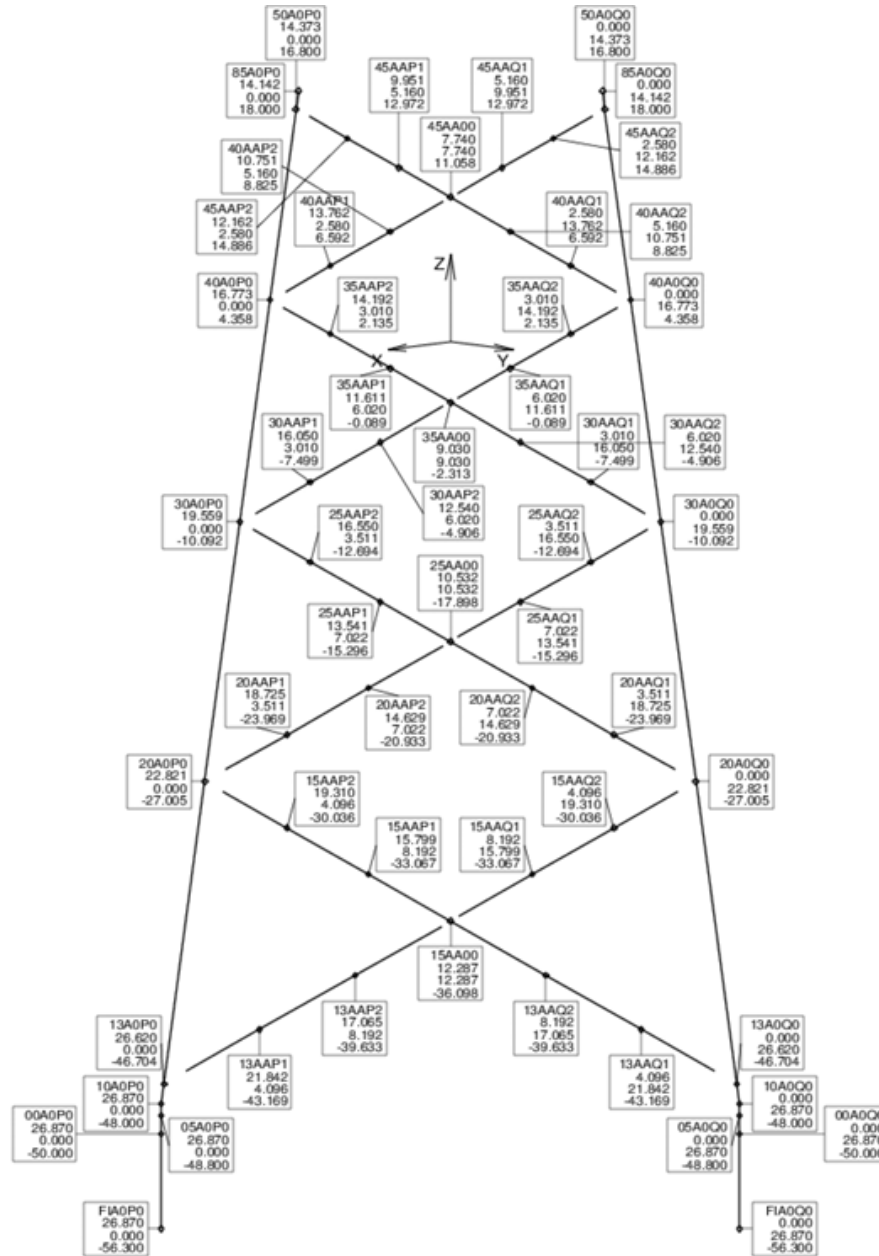


Figure B.1: Jacket Node placements for Initial Design. Pontow, S.,Kaufer, D., Shirzadeh, R. & Kühn M. (2017), *Design Solution for a Support Structure Concept for future 20MW*, IN-NWIND.EU, p. 67

C List of Mode Shapes

2D Truss			2D Frame			3D Truss			3D Frame		
Mode	[Hz]	Classification	[Hz]	Classification	Mode	[Hz]	Classification	[Hz]	Classification		
1	2.9	Global Bending 1	2.93	Global Bending 1	1	3.48	Global Bending 1	2.27	Asymmetric Brace 1		
2	9.07	Global Axial 1	9.03	Global Axial 1	2	3.48	Global Bending 1	3.24	Symmetric Brace 1		
3	10.54	Global Bending 2	10.32	Global Bending 2	3	5.34	Global Torsional 1	3.24	Symmetric Brace 1		
4	11.49	Transverse Leg 1	13.08	Transverse Leg 1	4	8.93	-	3.5	Global Bending 1		
5	18.42	Transverse Leg 2	15.38	Node Rotation 1	5	9.69	Global Bending 2	3.5	Global Bending 1		
6	18.57	Axial Leg 1	16.65	Global Bending 3	6	9.69	Global Bending 2	3.65	Asymmetric Brace 2		
7	20.88	Global Bending 3	17.5	Transverse Leg 2	7	10.38	-	4.42	Asymmetric Brace 3		
8	25.03	Breathing 1	17.88	Node Rotation 2	8	10.38	-	4.53	Symmetric Brace 2		
9	35.14	Global Bending 4	18.77	Axial Leg 1	9	10.45	Global Axial 1	4.53	Symmetric Brace 2		
10	35.53	Breathing 2	20.71	Global Bending 4	10	11.28	-	5.03	Asymmetric Brace 4		
11	42.82	Transverse Leg 3	21.46	Local Braces 1	11	11.63	-	5.06	Asymmetric Brace 5		
12	47.1	Axial Leg 2	23.89	Breathing 1	12	12.44	Global Torsional 2	5.36	Global Torsional 1		
					13	17.11	Global Bending 3	5.88	Symmetric Brace 3		
					14	17.11	Global Bending 3	5.88	Symmetric Brace 3		
					15	17.32	-	6.77	Asymmetric Brace 6		

Figure C.1: List of first 15 modes (2D) and 20 modes (3D) for the jacket-transition piece structure with corresponding natural frequencies.

Mode	2D Truss		2D Frame		3D Truss		3D Frame		3D Tower	
	[Hz]	Classification	[Hz]	Classification	[Hz]	Classification	[Hz]	Classification	[Hz]	Classification
1	0,182	Global Bending 1 (T)	0,181	Global Bending 1 (T)	0,192	Global Bending 1 (T)	0,19	Global Bending 1 (T)	0,2	Bending 1
2	0,596	Global Bending 2 (T)	0,594	Global Bending 2 (T)	0,192	Global Bending 1 (T)	0,19	Global Bending 1 (T)	0,2	Bending 1
3	1,84	Global Bending 3 (J/T)	1,86	Global Bending 3 (J/T)	0,308	Global Torsional 1	0,297	Global Torsional 1	0,32	Torsional 1
4	2,95	Global Axial 1	2,92	Global Axial 1	0,622	Global Bending 2 (T)	0,62	Global Bending 2 (T)	0,66	Bending 2
5	3,58	Global Bending 4 (J/T)	3,59	Global Bending 4 (J/T)	0,622	Global Bending 2 (T)	0,62	Global Bending 2 (T)	0,66	Bending 2
6	7,91	Global Bending 5 (J/T)	7,86	Global Bending 5 (J/T)	2,21	Global Bending 3 (J/T)	2,22	Global Bending 3 (J/T)	3,22	Bending 3
7	9,26	Global Axial 2	9,25	Global Axial 2	2,21	Global Bending 3 (J/T)	2,22	Global Bending 3 (J/T)	3,22	Bending 3
8	11	Global Bending 6 (J)	10,87	Global Bending 6 (J)	3,5	Global Axial 1	2,27	Asymmetric Brace 1	4,52	Axial 1
9	11,49	Transverse Leg 1	13,07	Transverse Leg 1	3,83	Global Bending 4 (J/T)	3,29	Symmetric Brace 1	8,57	Bending 4
10	15,53	Global Bending 7 (J/T)	15,19	Global Bending 7 (J/T)	3,83	Global Bending 4 (J/T)	3,29	Symmetric Brace 1	8,57	Bending 4
11	18,33	Axial Leg 1	15,58	Global Bending 8 (T)	5,14	Global Torsional 2	3,47	Global Axial 1	11,49	Torsional 2
12	20,76	Transverse Leg 2	17,05	Global Bending (J)	7,81	Global Bending	3,65	Asymmetric Brace 2	16,67	Bending 5
13	22,78	Global Axial 3	17,43	Transverse Leg 2	7,81	Global Bending	3,77	Global Bending 4 (J/T)	16,67	Bending 5
14	23,53	Global Bending 8 (T)	17,88	Node Rotation	8,93	-	3,77	Global Bending 4 (J/T)	18,96	Axial 2
15	25,19	Axial Leg 2	20,52	Global Bending (J)	9,88	Global Axial 2	4,42	Asymmetric Brace 3	23,2	Torsional 3

Figure C.2: List of the first 20 modes and corresponding natural frequencies for the combined structure, including the jacket, transition piece, tower, and RNA.

D Digital Appendix

The Virtual Appendix includes the structural dynamic analysis programs (D.1), the results and mode shape plots (D.2), and validation files (D.3).

D.1 Programs

The four analysis programs are provided in individual folders named:

- D11_VIB_2DTruss
- D12_VIB_2DFrame
- D13_VIB_3DTruss
- D14_VIB_3DFrame

Each program contains Python scripts for structural modal analysis using bar or beam elements in 2D or 3D, with built-in plotting and frequency extraction.

D.2 Mode Shape Plots

Mode shape plots resulting from the dynamic analysis are stored in a separate folder, organized by program type:

- D21_VIB_2DTruss
- D22_VIB_2DFrame
- D23_VIB_3DTruss
- D24_VIB_3DFrame

Each folder includes subfolders for plots of specific models.

Example folder structure for D24_VIB_3DFrame:

- D24_VIB_3DFrame:
 - D241_Jacket_Frequencies
 - D242_JacketTower_Frequencies
 - D243_Tower_Frequencies
- (Only present in Frame folders)

Each plot filename follows the structure

[ModelType]_[Component]_[modeXX]_[View].png

Where:

- **ModelType** – e.g., 3D_Frame
- **Component** – Jacket, JacketTower, or Tower
- **modeXX** – Mode number (0-based index)
- **View** – Viewpoint (e.g., XY, XZ), only used in 3D models

D.3 Validation Files

Validation files are included to support verification and correctness of the developed models and implementation. The following files are available:

- `D31_2D_Truss_validation.py`
- `D32_2D_Frame_validation.py`
- `D33_3D_Truss_validation.py`
- `D34_3D_Frame_validation.py`

These files validate the numerical implementation and confirm that the simulation results align with analytical solutions.

E Use of AI Technology

The following AI technology has been used.

GPT 4o:

Used for LaTeX syntax for tables, figures and references. Also, used for synonyms / word alternatives.

DeepL:

Used for grammar and spelling corrections. Also, used for synonyms / word alternatives.

GitHub Copilot:

Interactive AI integrated in Visual Studio Code. Supporting syntax errors and providing code suggestions. Also, used for debugging.

Microsoft Copilot:

Used for syntax for LaTeX for tables, figures and references. Also, used for code debugging, synonyms / word alternatives and translations.

Technical
University of
Denmark

Koppels Allé, Building 404
2800 Kgs. Lyngby

www.construct.dtu.dk