Liège University – Faculty of Applied Sciences

Parallel Harmonic Balance Method for the Analysis of Nonlinear Mechanical Systems

Victor Renkin

Thesis supervisor: Loïc SALLES
Lecturer, University of Liège

Jury members: Loïc SALLES
Lecturer, University of Liège
Romain BOMAN
Associate Professor, University of Liège
Ghislain RAZE
Associate Professor, University of Liège
Jie YUAN
Assistant Professor, University of Southampton
Alexandre HALBACH
CTO, Quanscient Oy

Master thesis presented in partial fulfillment
of the requirements for the degree of
Master of Science in Aerospace Engineering.

Academic year : 2025-2026

Acknowledgements

Throughout my years of study, I have been fortunate to receive guidance and support from many individuals, to whom I would like to express my sincere gratitude.

I would like to express my deepest gratitude to Loïc Salles for his time, guidance, and scientific support, all of which were essential to the completion of this work. His constant availability during more challenging moments and his unwavering support throughout this journey are also gratefully acknowledged. I am equally thankful to Alexandre Halbach for his guidance in the use of Quanscient and for patiently answering my numerous questions.

I sincerely thank Jie Yuan for offering me the opportunity to undertake an internship in Southampton. This experience proved to be highly enriching, both from a scientific and a personal perspective.

I am also grateful to Jiří Blahoš for sharing and communicating his results, which provided valuable insights and significantly contributed to the progress of this research.

I would like to express my heartfelt gratitude to my parents, who have consistently supported and encouraged me throughout my studies. I also wish to thank my partner's family for their kindness and support. A special mention is dedicated to Louise, whose constant presence, understanding, and encouragement made this journey considerably easier.

Finally, I would like to thank my friends for their encouragement and companionship during this period, in particular Corentin, Martin, Louis, and Lucas for their help in reviewing my work. I am especially indebted to Edwin, who devoted a significant amount of time to supporting me.

ABSTRACT

The accurate prediction of nonlinear vibration phenomena remains a key challenge in modern turbomachinery and aerospace design, as lightweight and highly integrated structures are increasingly affected by geometric nonlinearities, modal interactions, and amplitude-dependent dynamics that cannot be captured by classical linear vibration models. Robust frequency-domain formulations combined with scalable numerical solvers are therefore required to address industrial-scale finite element analyses.

This work presents a parallel Harmonic Balance framework for the analysis of geometrically nonlinear elastic systems. Large deformations are modelled using a finite element formulation based on the Green-Lagrange strain tensor. The application of the Harmonic Balance Method leads to a nonlinear algebraic system solved using a Newton–Raphson algorithm embedded within a continuation strategy, allowing nonlinear frequency responses (NLFRs) and nonlinear normal modes (NNMs) to be investigated in a unified manner. For the computation of nonlinear normal modes, an additional phase condition and a relaxation parameter are introduced to ensure uniqueness and convergence.

In parallel, a rotating-frame formulation is developed and integrated into the framework to account for rotational effects, enabling the computation of Campbell diagrams as well as their consistent extension to the evaluation of NNMs and NLFRs.

The framework is implemented in the in-house PYHARM solver and subsequently deployed on Quanscient's cloud-native multiphysics platform ALLSOLVE, which relies on domain decomposition and distributed-memory linear solvers (MUMPS, PETSC) for large-scale parallel finite element simulations. Comprehensive numerical studies are carried out on academic benchmark structures as well as on an industrial fan-blade configuration. Nonlinear frequency responses, nonlinear normal modes, and Campbell diagrams are analysed, and an excellent agreement between the different descriptions is observed, confirming the physical consistency of the proposed methodology. Scalability analyses demonstrate efficient strong scaling behaviour up to 64 MPI processes, with computational performance primarily governed by matrix assembly and factorisation. The Alternating Frequency–Time formulation is compared with a direct frequency-domain evaluation of the nonlinear terms, and the impact of quadruple-precision arithmetic is investigated, highlighting a trade-off between numerical robustness and computational cost.

Overall, the proposed framework enables predictive and scalable nonlinear vibration analysis in the frequency domain, while also providing a consistent nonlinear rotating-frame formulation for the investigation of rotating aerospace structures.

Keywords: Harmonic Balance Method, Nonlinear Frequency Response, Nonlinear Normal Modes, Geometric Nonlinearities, Continuation, Rotating Frame, Campbell Diagram, High-Performance Computing, QUANSCIENT ALLSOLVE.

This page intentionally left blank.

AI Acknowledgement

Artificial Intelligence (AI), specifically OpenAI's ChatGPT (`https://chatgpt.com`), was used throughout this work as a support tool. Its contribution focused primarily on refining the academic tone, improving linguistic accuracy, assisting with grammar, punctuation, and vocabulary, as well as providing support for LaTeX formatting.

AI was also used to enhance clarity and consistency in the written presentation. Its role was supportive rather than substitutive: the ideas, analyses, and scientific reasoning remain the sole responsibility of the author.

The use of AI aimed to improve the readability and coherence of the manuscript, while facilitating a more efficient writing process.

This page intentionally left blank.

Contents

Nomenclature **xi**

List of Acronyms **xv**

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Objectives . . . 3
- 1.3 Outline . . . 4

2 State of the Art **5**
- 2.1 Type of Nonlinearity . . . 5
- 2.2 Nonlinear Analysis . . . 6
  - 2.2.1 Nonlinear Frequency Response . . . 7
  - 2.2.2 Nonlinear Normal Mode . . . 8
- 2.3 Harmonic Balance Method . . . 9
- 2.4 Numerical Continuation . . . 10
  - 2.4.1 Numerous Software Packages . . . 10
- 2.5 Linear and Nonlinear Solvers . . . 11
  - 2.5.1 Parallel Linear Solvers . . . 11
  - 2.5.2 Domain Decomposition Methods . . . 12
  - 2.5.3 Parallelisation of the Harmonic Balance Method . . . 13
- 2.6 Conclusion . . . 13

3 Methodology **15**
- 3.1 Problem modelling . . . 15
  - 3.1.1 Deformation . . . 16
  - 3.1.2 Boundary Conditions . . . 17
  - 3.1.3 Finite Elements . . . 18
  - 3.1.4 Rotating Frame . . . 20
    - 3.1.4.1 Campbell diagram . . . 21
- 3.2 Harmonic Balance Method . . . 22
  - 3.2.1 Alternating Frequency-Time Method . . . 25
- 3.3 Nonlinear Solver . . . 25
  - 3.3.1 Bordering Algorithm . . . 27
- 3.4 Continuation . . . 28
  - 3.4.1 Regular Solution . . . 28
  - 3.4.2 Predictor–Corrector Schemes . . . 28
  - 3.4.3 Natural Parameter Continuation . . . 28

3.4.4 Advanced Predictors and Correctors 29
3.4.4.1 Tangent Predictor 29
3.4.4.2 Secant Predictor 30
3.4.4.3 Pseudo-Arclength Method 31
3.4.4.4 Arclength Method 33
3.4.5 Step Size Control 33
3.4.5.1 Iteration-Based Strategy 34
3.4.5.2 Angle-Based Strategy 34
3.4.6 Continuation Loop 36
3.5 Non-linear Normal Mode 37
3.5.1 Phase Condition 38
3.5.1.1 Simple Phase Condition 39
3.5.1.2 Robust Phase Condition 39
3.5.2 Equation of Motion with Fictive Energy 39
3.5.3 Continuation Scheme 40
3.5.3.1 Initialisation 40
3.5.3.2 Predictor 41
3.5.3.3 Corrector 41
3.6 QUANSCIENT ALLSOLVE Software 42
3.6.1 Source Code 42
3.6.2 Usage 43
3.6.2.1 Initialisation 43
3.6.2.2 Define the Mesh 43
3.6.2.3 Define the Field 43
3.6.2.4 Define the problem 44
3.6.2.5 Generating the Solver Quantities 44
3.7 PYHARM — Harmonic Balance Method Based Solver 45
3.7.1 Architecture and Functionality 45
3.7.2 Contributions of This Work 46
3.8 Conclusion 46

4 Results – PYHARM **49**
4.1 Single Degree of Freedom System 49
4.2 Two Degrees of Freedom System 50
4.3 Conclusion 51

5 Results – QUANSCIENT ALLSOLVE **53**
5.1 Hardware Overview 53
5.2 Testing Case 54
5.2.1 Clamped–Clamped Beam 54
5.2.1.1 Mode Shapes and Eigenfrequencies 55
5.2.1.2 Nonlinear Frequency Response 55
5.2.1.3 Nonlinear Frequency Response vs Backbone Curve 56
5.2.2 Cantilever beam 57
5.2.2.1 Mode Shapes and Eigenfrequencies 58
5.2.2.2 Nonlinear Frequency Response 58
5.2.2.3 Nonlinear Frequency Response vs Backbone Curve 60

5.3 Comparison of Continuation Strategies 60
    5.3.1 Comparison of Step-Size Methods 61
    5.3.2 Comparison of Predictor–Corrector Methods 62
5.4 Validation of the Rotating-Frame Formulation 62
    5.4.1 Verification Through Axial Deformation 63
    5.4.2 Verification Through Campbell Diagram 64
    5.4.3 Nonlinear Normal Mode Verification 65
    5.4.4 Nonlinear Frequency Response Analysis 66
5.5 Scalability 67
    5.5.1 Reference configuration for scalability tests 68
    5.5.2 Strong Scalability Small Testcase 68
        5.5.2.1 Alternating Frequency-Time and Analytical Formulations . . 70
        5.5.2.2 Scalability of the Solver in Quadruple Precision 71
    5.5.3 Strong Scalability Large Testcase 73
5.6 Industrial Case 74
    5.6.1 Mode Shapes and Eigenfrequencies 75
    5.6.2 Nonlinear Frequency Response with Backbones 75
    5.6.3 Campbell Diagram 76
5.7 Conclusion 77

6 General Conclusion and Perspectives **80**
6.1 Summary of Achievements 80
6.2 Limitations 81
6.3 Future Work 82

Bibliography **84**

This page intentionally left blank.

Nomenclature

α, β	Rayleigh damping coefficients for mass and stiffness matrices, respectively
$\boldsymbol{\Omega}$	Skew-symmetric angular velocity tensor associated with the rotating reference frame
$\boldsymbol{\sigma}$	Cauchy stress tensor (Pa)
$\boldsymbol{a}(\boldsymbol{x}^0)$, $\boldsymbol{b}(\boldsymbol{x}^0)$, $\boldsymbol{c}(\boldsymbol{x}^0)$	Prescribed vector fields on the boundary $\partial\boldsymbol{\Omega}_0$, associated respectively with Dirichlet, Neumann, and Robin conditions.
$\boldsymbol{c}_i$	Connection vector defined as $\boldsymbol{c}_i = (\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\, \omega_i - \omega_{i-1})$, combining displacement and frequency increments between two successive solution points
$\boldsymbol{D}$	Damping matrix (kg/s)
$\boldsymbol{E}, \varepsilon$	Strain tensor nonlinear and linear(–)
$\boldsymbol{f}_\Omega$	Rotation-induced centrifugal load vector
$\boldsymbol{f}_{ext}(\omega, t)$	External excitation force vector (N)
$\boldsymbol{f}_{nl}(\boldsymbol{u}, \dot{\boldsymbol{u}})$	Nonlinear internal force vector (N)
$\boldsymbol{K}$	Stiffness matrix (N/m)
$\boldsymbol{K}_\Omega$	Rotation-induced centrifugal softening stiffness matrix
$\boldsymbol{M}$	Mass matrix (kg)
$\boldsymbol{r}$	Spatial position vector of a material point expressed in the rotating frame
$\boldsymbol{S}$	Piola-Kirchhoff stress tensor (Pa)
$\boldsymbol{u}, U$	Displacement vector field (m)
$\boldsymbol{u}^k$, ω^k, μ^k	Values of displacement, frequency, and relaxation parameter at a Newton iteration
$\boldsymbol{u}^p$, ω^p, μ^p	Predictor values of displacement, frequency, and relaxation parameter
$\boldsymbol{x}^0, \boldsymbol{x}$	Spatial coordinates in the reference and current configurations, respectively (m)
$\boldsymbol{y}$	Vector gathering all components to be solved in the problem

$\ddot{\boldsymbol{u}}$	Acceleration vector field $(\mathrm{m/s^2})$
$\delta \boldsymbol{E}^j$	Variation of the Green–Lagrange strain tensor with respect to a test function
$\dot{\boldsymbol{u}}$	Velocity vector field $(\mathrm{m/s})$
$\Gamma_D, \Gamma_N, \Gamma_C, \Gamma_R$	Portions of the boundary $\partial\boldsymbol{\Omega}_0$ where Dirichlet, Neumann, Cauchy, and Robin conditions are applied, respectively
γ_i	Angle between the connection vector $\boldsymbol{c}_i$ and the tangent vector $\boldsymbol{\tau}_i$
γ_{opt}	Prescribed optimal angle used as a reference in the adaptive step size strategy
$\mathbb{R}$	Set of real numbers
$\mathbf{f_0}, \boldsymbol{f}$	External force field in the undeformed and deformed coordinate systems, respectively
$\mathbf{f}$	Combination of $\boldsymbol{f}_{\mathrm{ext}}$ and $\boldsymbol{f}_{\mathrm{nl}}$ in the time domain
$\mathbf{G}$	General set of nonlinear algebraic equations representing the residual used in the HBM system
$\mathbf{Z}$	Matrix of size $(2N_H+1)n \times (2N_H+1)n$ describing the linear dynamics in the harmonic balance formulation
$\mathcal{B}_0, \mathcal{B}$	Reference and current configurations of the body in the undeformed and deformed states, respectively
μ	Relaxation parameter
∇	Vector differential operator
$\nabla_{\tilde{\boldsymbol{u}}} \boldsymbol{G}$	Jacobian matrix of $\boldsymbol{G}$ with respect to the frequency-domain solution vector $\tilde{\boldsymbol{u}}$
ω	Base frequency of vibration
ψ	Sensitivity parameter controlling the influence of the angle γ_i on the scaling factor
ρ_0, ρ	Material density in the undeformed and deformed coordinate systems, respectively
τ	Prediction vector in the continuation procedure
$\tilde{\varphi}_i$	Null-space eigenvector of the Jacobian, associated with the phase invariance of the periodic solution
$\tilde{e}_{\mathrm{fic}}$	Fictive energy
$\tilde{\boldsymbol{f}}_0, \tilde{\boldsymbol{f}}^c, \tilde{\boldsymbol{f}}^s$	Constant, cosine, and sine components of $\tilde{\boldsymbol{f}}$
$\tilde{\boldsymbol{u}}$	Vector of harmonic (Fourier) coefficients for displacement in the frequency domain
$\tilde{\boldsymbol{u}}_0, \tilde{\boldsymbol{u}}^c, \tilde{\boldsymbol{u}}^s$	Constant, cosine, and sine components of $\tilde{\boldsymbol{u}}$
$\tilde{\mathbf{f}}$	Combination of $\tilde{\boldsymbol{f}}_{\mathrm{ext}}$ and $\tilde{\boldsymbol{f}}_{\mathrm{nl}}$ in the frequency domain

$\tilde{\nabla}$	Block-diagonal differentiation operator in the frequency domain, with each block $\tilde{\nabla}_k$ corresponding to the derivative operator for the k-th harmonic
ϖ	Angular speed of the rotating structure
ξ_i	Scaling factor applied at step i to update the continuation step size
C	Fourth-order stiffness tensor written as a matrix (Pa)
g	Additional constraint function in the correction stage of the continuation procedure
n_H	Number of harmonics
N_{max}	Maximum allowed number of Newton iterations
N_i	Number of Newton iterations performed at step i
s	Size of the prediction step in the continuation procedure
$T_{\boldsymbol{u}}, T_{\tilde{\boldsymbol{u}}}$	Operator of translation in the temporal and frequency domains
v	Finite element test functions

List of Acronyms

AFT Alternating Frequency–Time.
AWS Amazon Web Services.

DDM Domain Decomposition Method.
DOF Degree-of-Freedom.

E Edge-wise Bending Mode.

F Flap-wise Bending Mode.
FEM Finite Element Method.
FFT Fast Fourier Transform.

HBM Harmonic Balance Method.
HEX Hexahedral element.

IFFT Inverse Fast Fourier Transform.

MEMS Micro-Electro-Mechanical Systems.
MPI Message Passing Interface.

NLFR Nonlinear Frequency Response.
NNM Nonlinear Normal Mode.

T Torsion Mode.
TET Tetrahedral element.

1

INTRODUCTION

1.1 Context and Motivation

Vibration is a fundamental and omnipresent phenomenon in mechanical systems. It arises across all scales, from the subtle oscillations in Micro-Electro-Mechanical Systems (MEMS) [1] devices to the large-scale deformations of aerospace structures such as rotating blades and fan assemblies [2]. Whenever a structure is exposed to periodic excitation whether from rotational imbalance, aerodynamic loading, or ambient disturbances, vibrations inevitably occur. If uncontrolled, they can lead to accelerated wear, reduced precision, or even catastrophic failure. Even small amplitude vibrations, when sustained over long durations, can induce fatigue damage. Consequently, the ability to model, predict, and mitigate vibration is essential in mechanical design. Understanding how and why a structure vibrates is not merely diagnostic; it enables performance optimisation, improves reliability, and extends operational lifespan. Vibration analysis thus lies at the heart of resilient and efficient mechanical engineering.

As modern systems grow increasingly lightweight, complex, and multifunctional, the assumptions underpinning linear vibration theory begin to break down. In aerospace and other high-performance applications, advanced materials, intricate assemblies, and tight integration introduce nonlinear dynamic effects that cannot be ignored [3]. Phenomena such as contact friction [4], geometric nonlinearity [5, 6] become intrinsic to the system's behaviour. Classical modal analysis, which assumes linearity, often fails to capture these effects. Nonlinear vibrations introduce features like amplitude dependent frequencies [7], internal resonances [8], and multi-mode energy exchanges between modes features that demand more sophisticated analytical tools [9]. Addressing this complexity is not optional; it is necessary for achieving accurate, safe, and robust designs.

Among the approaches [10] for solving nonlinear bondary value problem, the Harmonic Balance Method (HBM) provides a powerful frequency-domain framework [11] for the analysis of nonlinear vibration phenomena. By approximating the periodic response with a truncated Fourier series, the governing equations are recast into a set of nonlinear algebraic equations [12]. Continuation techniques, such as the pseudo-arclength predictor–corrector approach, are then employed to track the evolution of both Nonlinear Frequency Responses (NLFRs) and Nonlinear Normal Modes (NNMs) along their respective curves [11, 13]. This combination offers a consistent and systematic methodology for exploring the dynamic behaviour of nonlinear systems. This approach offers significant advantages for large-scale problems, often outperforming brute-force time integration in efficiency. HBM has been successfully applied to nonlinear vibration analysis of mechanical structures, from academic benchmarks [14] to realistic components [15].

However, most applications of the HBM remain constrained by the number of harmonics included in the formulation. Increasing the harmonic content improves accuracy by capturing higher frequency components and sharper nonlinear effects, but also leads to a substantial rise in the number of unknowns and a denser, more computationally expensive system. This trade-off becomes particularly restrictive in large-scale simulations. In practical industrial settings such as aerospace and turbomachinery, full finite element models routinely involve several million Degree-of-Freedoms (DOFs) [16, 17]. This level of complexity is illustrated in Figure 1.1.1, which shows a large whole-engine model comprising approximately 34 millions DOFs, representative of contemporary high-fidelity simulations used in realistic engineering applications.

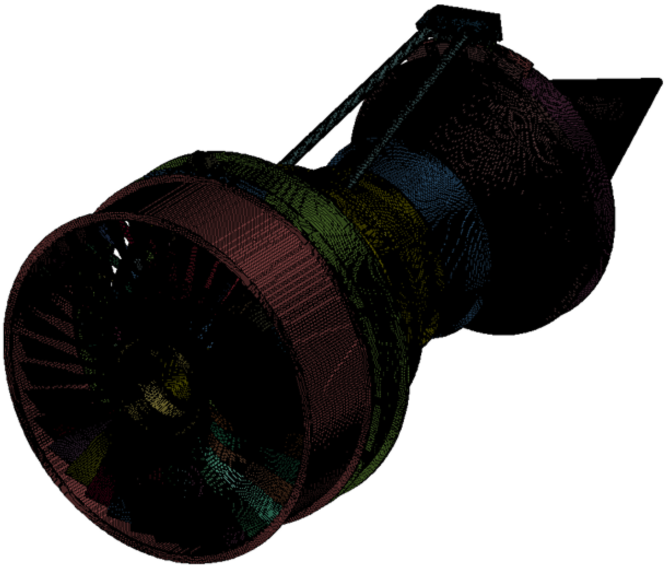


Figure 1.1.1: Large whole engine model with 34 million DOFs [17].

While linear modal analysis is routinely applied to such systems, whereas their nonlinear counterparts remain computationally demanding. Distributed nonlinearities introduce significant stiffness and strong harmonic coupling, requiring fine spectral resolution and iterative solvers that scale poorly with increasing harmonic order. As a result, classical implementations of the HBM often become impractical beyond a limited number of harmonics, thereby restricting their applicability to industrial-scale models. Recent advances in high-performance computing and parallel numerical libraries have nevertheless opened new perspectives for overcoming these limitations [13]. The present work follows this direction by extending nonlinear frequency-domain analysis to large-scale systems through the integration of modern continuation algorithms with scalable, distributed computation.

Within this framework, attention is paid to structural rotation, which is highly relevant in practical applications such as turbomachinery [18, 19]. Rotation induces centrifugal effects that modify the effective stiffness and vibration characteristics of the structure, affecting resonance conditions and the evolution of natural frequencies with rotational speed, as commonly illustrated by Campbell diagrams. Accounting for these effects is therefore essential for obtaining physically meaningful vibration predictions.

1.2 Objectives

While continuation methods have long been established in the analysis of nonlinear dynamical systems, their application to large-scale industrial models has been limited by significant computational demands. Recent advances in parallel computing and high-performance solver infrastructures now offer the opportunity to revisit these techniques in the context of high-fidelity simulations.

The objective of this thesis is to establish a general continuation framework interfaced with Quanscient Allsolve, a high-performance finite element solver supporting parallel computing and direct interaction through Python scripting. The framework is formulated for problems governed by the equations of motion of nonlinear structural dynamics and is intended for the computation of both NLFRs and NNMs using the HBM. For the computation of NNMs, specific phase conditions and relaxation strategies are introduced to ensure robustness and uniqueness of the solutions. Several predictor–corrector schemes are investigated, including tangent- and secant-based predictors combined with pseudo-arclength or classical arclength correctors.

In addition to purely translational dynamics, an additional objective of this work is to extend the modelling framework to rotating structural systems. This extension focuses on the inclusion of rotational effects such as centrifugal stiffening, gyroscopic contributions, and the resulting stiffness softening behaviour, which are essential for realistic turbomachinery applications. The consideration of these effects enables the analysis of vibration characteristics as a function of the rotational speed and the construction of Campbell diagrams to identify critical speeds, frequency splitting, and modal interactions in rotating components. These rotational effects are also incorporated into the computation of the NNMs and the evaluation of nonlinear forced responses through the NLFRs.

A further objective is to assess the scalability of Quanscient Allsolve with increasing numbers of Message Passing Interface (MPI) processes, thereby quantifying the parallel efficiency of the proposed approach on large-scale finite element models. The overall research scope is summarised in Figure 1.2.1.

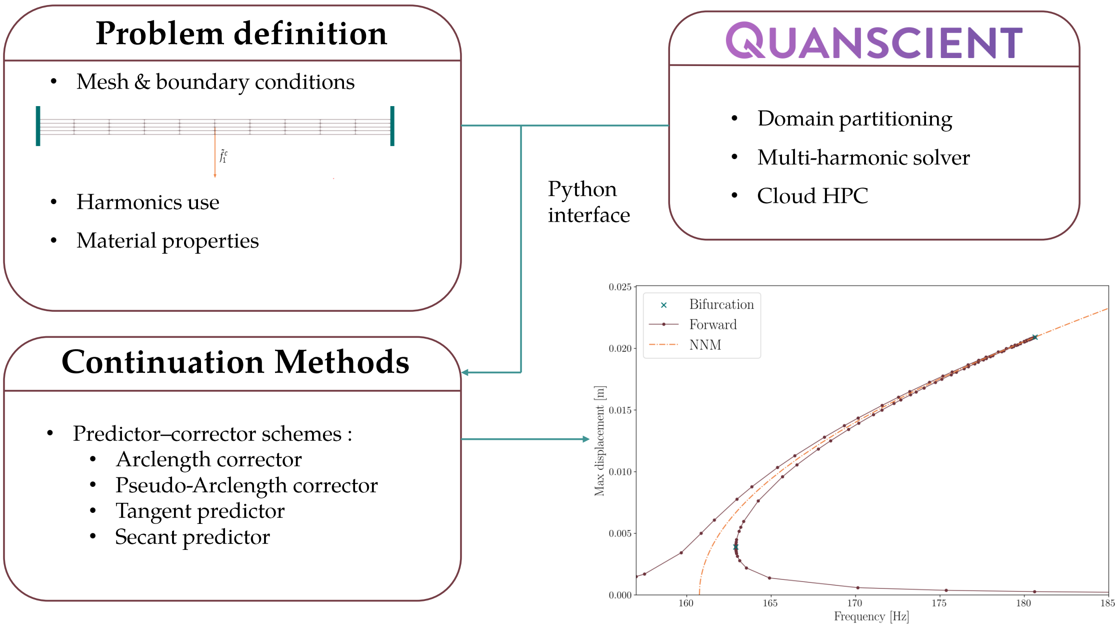


Figure 1.2.1: Conceptual diagram of the research objectives.

1.3 Outline

This thesis is divided into 6 chapters, each addressing a specific objective.

Chapter 2 reviews the state of the art in nonlinear vibration analysis. The main types of nonlinearities are discussed, nonlinear analysis concepts such as NLFRs and NNMs are introduced, and the HBM, numerical continuation techniques, as well as existing linear and nonlinear solver technologies are reviewed. This chapter provides the background and motivation for the methodological developments presented in this work.

Chapter 3 presents the methodology of this thesis. It begins with the physical modeling of the structure, followed by its finite element formulation, including the treatment of rotational effects, and its reformulation within the harmonic balance framework. Key numerical tools are then introduced, including linear modal analysis, the Alternating Frequency–Time (AFT) method, and continuation strategies, which enable systematic tracking of NLFRs and NNMs along their characteristic curves. The chapter concludes by introducing the computational platforms used: the dedicated Harmonic Balance code pyHarm, and the cloud-native multiphysics simulation platform Quanscient Allsolve.

Chapter 4 reports the validation of the methodology through the computation of NNMs with pyHarm. Academic benchmark cases are investigated to verify the correctness of the implementation and to illustrate the main capabilities of the developed framework.

Chapter 5 applies the continuation framework to the high-performance platform Quanscient Allsolve. Benchmark structures such as the clamped–clamped beam and the cantilever beam are examined alongside an industrial case study. The chapter evaluates scalability and compares different numerical strategies, thereby demonstrating the applicability of the methodology to large-scale problems.

Chapter 6 concludes the thesis by summarising the main findings, highlighting the limitations of the present work, and outlining possible directions for future research.

2

STATE OF THE ART

This chapter reviews the theoretical background, computational methods and existing applications relevant to the analysis of nonlinear mechanical vibrations. The discussion begins with an overview of the different types of nonlinearities in order to clarify their physical origin and implications for structural dynamics, as introduced in Section 2.1. Attention then turns to nonlinear analysis techniques in Section 2.2, where two complementary perspectives are introduced: the forced response of the system through NLFRs methods, and the investigation of intrinsic resonant mechanisms via NNMs.

The HBM is presented in Section 2.3 as a frequency-domain strategy for computing periodic solutions in nonlinear systems. A review of numerical continuation techniques is provided in Section 2.4, highlighting their role in systematically tracking solution branches under parameter variations.

The final part of this chapter, Section 2.5, addresses the linear and nonlinear solvers that underpin these computational frameworks, with emphasis on parallelisation strategies and Domain Decomposition Method (DDM). These existing approaches are examined to identify their limitations and to motivate the developments proposed in this thesis.

2.1 Type of Nonlinearity

Nonlinearities in mechanical systems can be either localised or distributed, depending on their physical origin and spatial extent. This distinction is particularly evident in aero-engines, where numerous nonlinear mechanisms coexist within a single assembly [17]. The schematic in Figure 2.1.1 illustrates typical sources of nonlinearity in such engines, including contact, friction, geometric effects, and material behaviour. Each of these mechanisms may influence the dynamic response differently, depending on whether they are confined to specific interfaces or distributed throughout the structure.

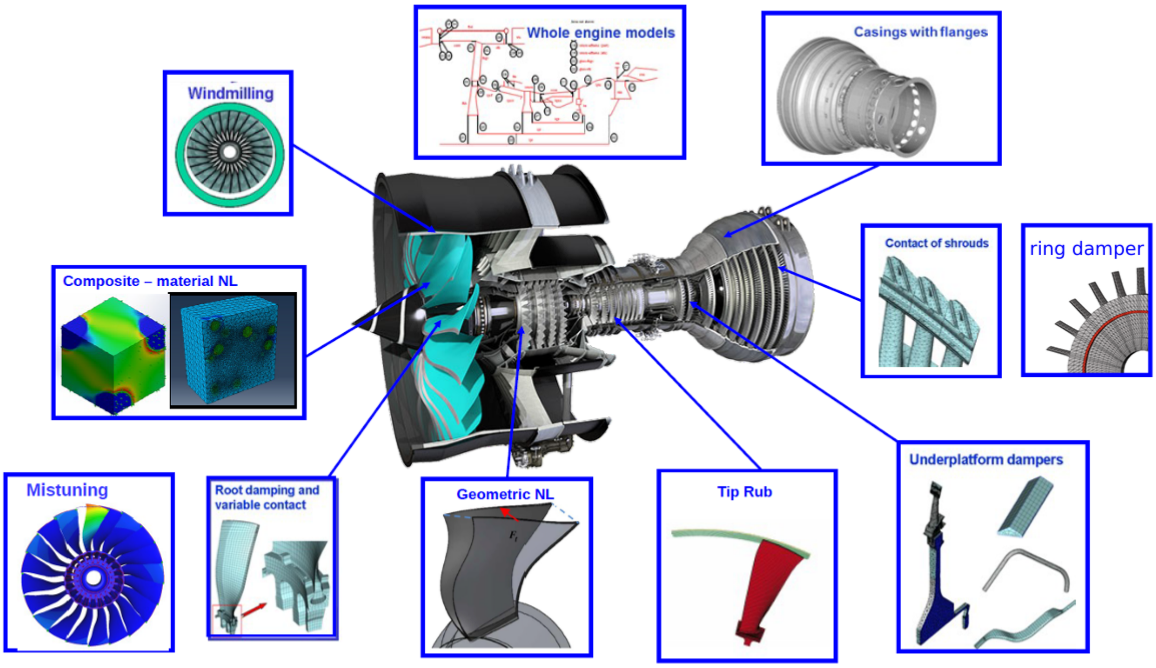


Figure 2.1.1: Typical sources of nonlinearity in an aero-engine [17].

Localised nonlinearities generally arise from frictional contact or mechanical joints [4], such as shroud interfaces, under-platform dampers, or blade roots. These phenomena are confined to a limited region of the structure and are often modelled using reduced-order approaches that preserve linear dynamics for the bulk structure while introducing nonlinear functions only at the interface nodes [20].

Distributed nonlinearities, on the other hand, emerge from material and geometric effects [21, 22]. Material nonlinearities appear in composites or metals exhibiting plasticity, damage, or anisotropy, while geometric nonlinearities result from large displacements, rotations, or deformations that alter the structural stiffness during motion. These mechanisms are not confined to a single interface but spread across the entire component or assembly, making them more challenging to reduce efficiently.

These nonlinear effects are taken into account through the analysis of NLFRs or NNMs, which form the basis of modern nonlinear analysis frameworks.

2.2 Nonlinear Analysis

The dynamic behaviour of nonlinear structures can no longer be fully described using the properties of linear normal modes, as the principle of superposition is no longer valid [23]. In such systems, stiffness and damping vary with the amplitude of motion, leading to energy-dependent dynamics that require dedicated analysis tools. Two complementary approaches are typically employed to characterise these behaviours: NLFRs, discussed in Section 2.2.1, and NNMs, presented in Section 2.2.2.

2.2.1 Nonlinear Frequency Response

NLFRs describe the steady-state behaviour of a structure subjected to a harmonic force of fixed amplitude while the excitation frequency is swept. The resulting curve represents the equilibrium amplitude of the periodic response for each excitation frequency. Unlike linear systems, where the frequency response is single-valued and symmetric, nonlinear systems may exhibit multiple coexisting steady-state solutions, amplitude-dependent resonance frequencies [7], sudden jumps [24], internal resonances [8], and multi-mode energy exchanges [9]. These phenomena are illustrated in Figure 2.2.1, which presents a representative example of a Duffing-type nonlinear oscillator.

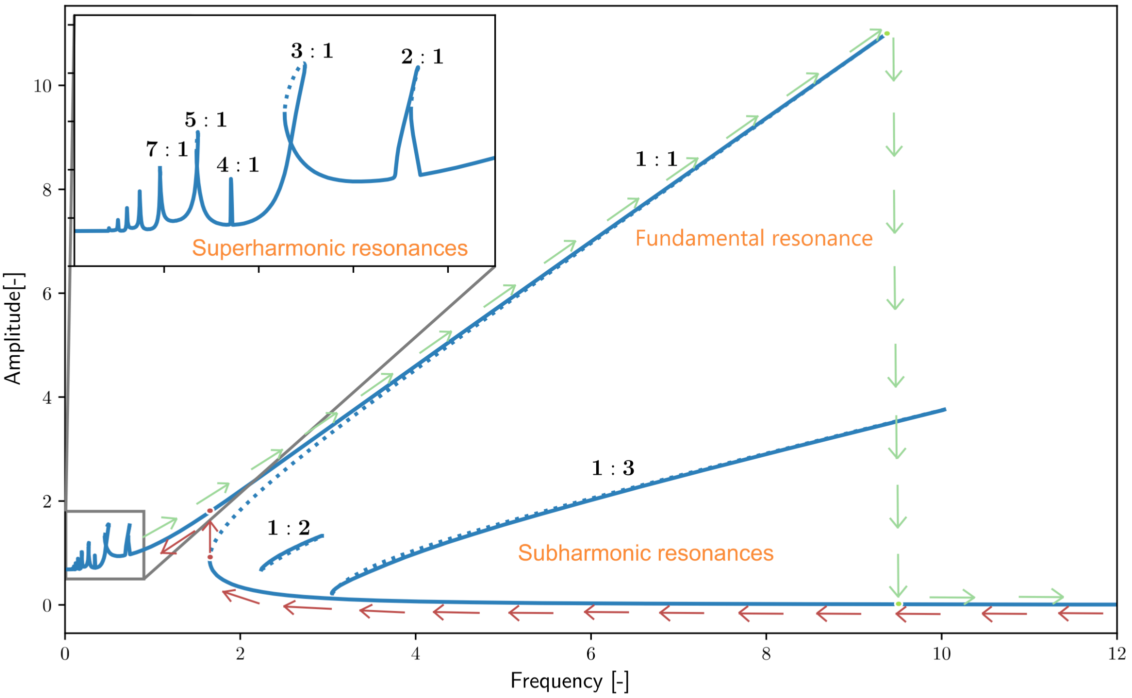


Figure 2.2.1: NLFR of a Duffing oscillator. The solid line represents the stable steady-state solutions, while the dashed line indicates the unstable branch. The green arrows (🟩) denote the forward frequency sweep, and the red arrows (🟥) correspond to the backward sweep [25].

Typical phenomena observed in NLFR include a bending of the resonance peak caused by stiffness nonlinearity, leading to either a hardening (right-leaning) or softening (left-leaning) behaviour. As the excitation frequency is gradually increased or decreased like a sweep frequency, the system may undergo abrupt transitions between stable and unstable branches, producing hysteresis and jump phenomena [3]. At higher forcing levels, additional features such as superharmonic and subharmonic resonances, internal modal interactions, or even quasiperiodic responses may appear [25]. These effects reveal the richness of nonlinear dynamics and the underlying energy transfer mechanisms between modes.

The main objective of nonlinear frequency response analysis is to capture how resonance characteristics evolve with excitation level, to predict large-amplitude vibrations, and to identify nonlinear features that cannot be detected using linear methods. In experimental testing, these curves are essential for defining safe operating ranges, validating nonlinear computational models, and improving reduced-order representations.

2.2.2 Nonlinear Normal Mode

Over the years, several definitions of NNMs have emerged, each providing a distinct but complementary perspective on how the concept of modes can be generalised to nonlinear systems. The first formal definition was proposed by Rosenberg [26–28], who described NNMs as periodic oscillations in which all coordinates of a system move synchronously. This intuitive definition provided a natural generalisation of linear modes to nonlinear settings and remains widely used for conservative systems. This notion was later broadened to include asynchronous periodic motions, thereby enabling the description of internal resonances and modal interactions [29].

A second and more general framework was later established by Shaw and Pierre [30], who defined NNMs as invariant manifolds embedded in the system's phase space. In this geometric interpretation, the motion of the nonlinear system is confined to a low-dimensional invariant surface that represents the nonlinear continuation of a linear mode. An example for a two DOFs system is illustrated in Figure 2.2.2, showing both in-phase and out-of-phase conditions [23].

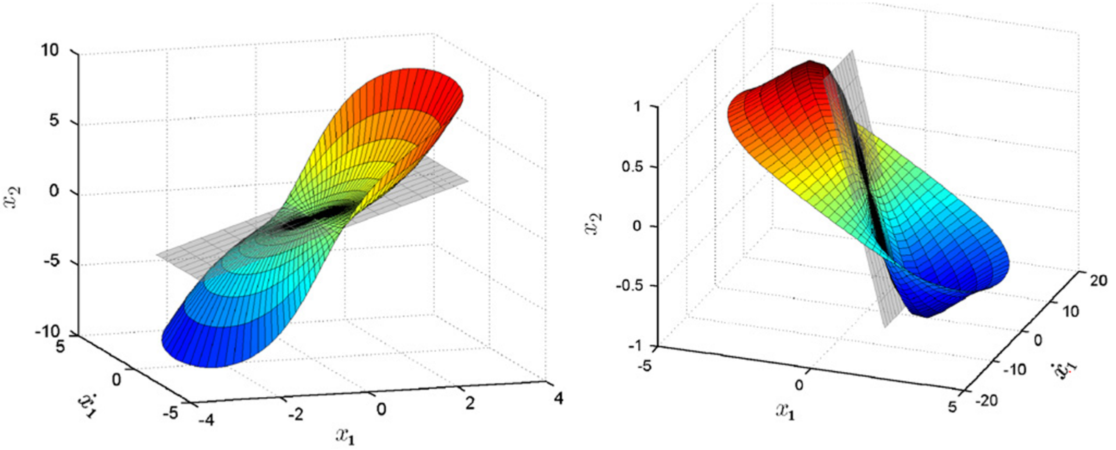


Figure 2.2.2: Two-dimensional invariant manifold of a two-DOFs system, with the corresponding linear normal modes in grey. Left: in-phase condition; right: out-of-phase condition [23].

This formulation provides a rigorous characterisation of the dynamics induced by geometric nonlinearities and forms a solid basis for invariant-manifold-based model reduction, where the governing equations are projected onto a low-dimensional nonlinear subspace to derive accurate reduced-order models [5, 6]. In this context, the open-source software `MORFEInvariantManifold.jl` implements the high-order direct parametrisation of invariant manifolds for geometrically nonlinear finite element structures, as detailed in [31, 32].

In this work, the extended definition of Rosenberg, which admits asynchronous periodic motions, is adopted. By relaxing the requirement of synchronous extrema among the system coordinates, this formulation provides a consistent and flexible description of periodic oscillations in nonlinear systems. Such a definition is essential for capturing the intrinsic dynamics induced by nonlinear effects, for which the notion of linear normal modes becomes inadequate.

Within this framework, NNMs emerge as the natural generalisation of linear modes to the nonlinear regime and form the basis for analysing how modal properties evolve with vibration amplitude. At low energy levels, the system response remains close to the linear behaviour, and resonance occurs when the excitation frequency coincides with the natural frequency of an NNM. As the energy stored in the system increases, nonlinear effects progressively modify both the oscillation frequency and the associated deformation pattern. This amplitude-dependent evolution is classically described through the backbone curve, which may exhibit hardening or softening behaviour depending on the nature of the nonlinearity [23].

The backbone curve, illustrated in Figure 2.2.3, represents this intrinsic link between frequency and amplitude in the absence of external forcing and damping [23]. It defines the fundamental dynamic signature of a nonlinear system, describing how its natural frequency evolves as the oscillation amplitude grows. In lightly damped structures, the backbone generally aligns with the resonance peaks of the NLFRs. Although dissipation and excitation can slightly shift these peaks, the backbone remains a reliable indicator of the underlying nonlinear dynamics.

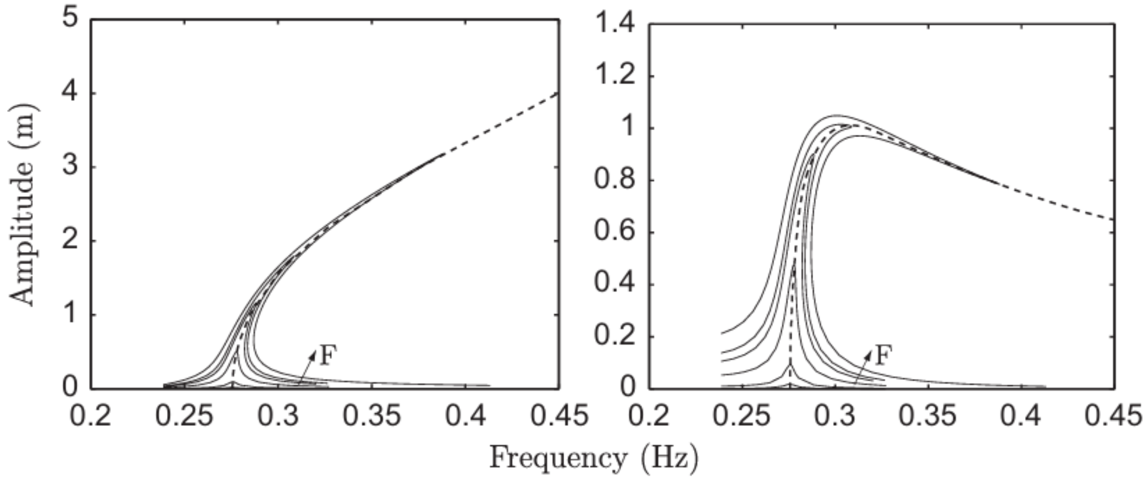


Figure 2.2.3: Backbone curves of a two DOFs initially at the same frequency, shown together with NLFR at different amplitudes, adapted from [23].

A complete state of the art of the NNMs is made on the PHD of Clement Grenat [1] which the information have this come from.

2.3 Harmonic Balance Method

Nonlinear mechanical systems exhibit a broad range of complex dynamical behaviours, as discussed in the previous sections. Accurately computing periodic solutions is essential for understanding these phenomena, since such responses often form the backbone of the system's attractors. The HBM has become a widely adopted approach for this purpose. It relies on the assumption that the solution of the nonlinear equations of motion is periodic and can be approximated by a truncated Fourier series [12]. By projecting the governing equations onto a Fourier basis through a Galerkin procedure, the original system of differential equations is transformed into a set of nonlinear algebraic equations expressed directly in terms of the harmonic coefficients [11]. For this reason, the method is also commonly referred to as the Fourier–Galerkin approach.

A major advantage of HBM lies in its efficiency when applied to systems with smooth nonlinearities, where only a limited number of harmonics is typically required to achieve high accuracy [11]. In contrast to time-domain techniques such as shooting or collocation, HBM offers a more computationally attractive alternative for large-scale problems, particularly when direct time integration becomes prohibitively expensive [33].

The effectiveness of HBM has also been demonstrated in large scale applications beyond structural dynamics. In electronics, it is a foundational tool for radio-frequency circuit steady-state analysis [34], where parallel harmonic balance algorithms enable the efficient simulation of complex microwave and analogue systems [35]. In turbomachinery computational fluid dynamics, harmonic balance formulations often referred to as nonlinear frequency-domain or time-spectral methods have become state-of-the-art for capturing periodic flow unsteadiness [36] at a fraction of the cost of full transient simulations [37]. These cross-disciplinary successes highlight the maturity and versatility of HBM and emphasise its strong potential for nonlinear vibration analysis, especially when enhanced through parallel computing.

A comprehensive state-of-the-art review of the HBM, including its variants, numerical challenges and modern extensions, is provided by Dai et al. [33], who offer an in-depth comparison of the different formulations and their respective domains of applicability.

2.4 Numerical Continuation

The continuation method is a cornerstone of nonlinear analysis, providing a systematic framework for tracing the solutions of parameter-dependent equations. Its origins can be traced back to the pioneering works of Poincaré [38], Klein [39], and Bernstein [40], who introduced early concepts related to the continuous variation of solutions with respect to system parameters in nonlinear problems.

The development of continuation techniques has evolved significantly over time, leading from simple natural continuation schemes to more advanced and robust strategies. Among these, predictor–corrector algorithms [41] have played a key role by enabling the systematic tracking of solution branches. These approaches combine a prediction step, often based on a local tangent approximation, with a correction step typically performed through Newton–Raphson iterations. The introduction of pseudo–arc length parametrisation further enhanced their capabilities by allowing the continuation to pass through singular points such as folds or turning points, thus ensuring reliable path-following even near bifurcations [42, 43]. A comparison between the various continuation strategies highlights the advantages of these more sophisticated methods in terms of robustness and convergence efficiency [44].

In parallel with predictor–corrector schemes, alternative continuation formulations were introduced to improve convergence and efficiency. Among these, the Asymptotic Numerical Method (ANM) proposed by Damil and Potier-Ferry [45] is based on an asymptotic perturbation framework, in which the solution is expressed as a high-order power series with respect to an artificial path parameter, thereby avoiding any predictor–corrector strategy or iterative correction procedure.

2.4.1 Numerous Software Packages

Several software tools exist for continuation and bifurcation analysis of nonlinear systems. The Table 2.4.1 summarises the main ones, their programming languages, and their main features.

Software	**Language**	**Main Features**
AUTO [46]	Fortran	Early and well-known continuation code for ODEs; supports OpenMP and MPI parallelisation.
MATCONT [47]	Matlab	Widely used for bifurcation analysis of ODEs.
MANLAB [48]	Matlab	Interactive continuation of nonlinear algebraic systems using harmonic balance.
COCO [49]	Matlab	Modular framework for continuation and bifurcation.
NLVIB [50]	Matlab	Focused on nonlinear vibration analysis of mechanical structures.
NI2D [51]	Matlab	Detects, visualises, and estimates structural nonlinearities.
LOCA [52]	C++	Part of Trilinos; scalable continuation and bifurcation library using OpenMP/MPI.

Table 2.4.1: Overview of widely used numerical continuation and bifurcation analysis software [13].

Efficient continuation relies on robust numerical solvers. Newton-based schemes often solve large linear systems, making solver choice and parallelisation crucial for performance and scalability.

2.5 Linear and Nonlinear Solvers

The information presented in this section and the following related ones is largely based on the doctoral thesis of Blahoš [13].

Solving systems of equations is often the most time-consuming part of the entire computational process. As a result, significant research has been devoted to the development of both nonlinear and linear solvers. These two classes of solvers are typically employed in a hierarchical fashion. An outer nonlinear solver iteratively calls an inner linear solver to resolve linearised systems until a sufficiently accurate approximation of the solution is reached. This is the case, for example, with Newton–Raphson methods.

Linear solvers are generally classified into two categories: direct and iterative methods. The principal distinction lies in their approach. Direct solvers compute the exact solution (up to machine precision) in a finite number of operations, typically through matrix factorisation. In contrast, iterative solvers begin with an initial guess and refine it through successive iterations until convergence criteria are satisfied. While iterative methods can be prematurely stopped to yield an acceptable approximate solution, this is not possible with direct methods. However, iterative solvers often exhibit greater sensitivity to ill-conditioned matrices [53]. In this thesis, only direct solvers are employed.

In large-scale problems, however, solver performance increasingly depends on the ability to exploit parallel computing resources. The following subsections address this aspect, beginning with parallel linear solvers in Section 2.5.1, then reviewing domain decomposition techniques in Section 2.5.2, and finally discussing previous attempts to parallelise the HBM in Section 2.5.3.

2.5.1 Parallel Linear Solvers

Parallelism is an essential component of modern linear solvers. The field of parallel computing emerged with the development of multiprocessor architectures and has gained importance as modern supercomputers now feature millions of processing cores [54]. To efficiently exploit such hardware (including clusters and cloud-based high-performance platforms), parallel algorithms divide the problem into independent tasks, each handled by separate threads or processes. Various solvers are listed in Table 2.5.1.

Software	**Type**	**Parallelism Model**
MUMPS [55]	Direct (LU, Cholesky, LDL^T)	MPI (distributed memory)
SUPERLU [56]	Direct (LU)	MPI + OpenMP (hybrid)
PARDISO [54]	Direct (LU, LDL^T, QR)	OpenMP / MPI
PETSC [57]	Iterative and direct	MPI / OpenMP (hybrid)

Table 2.5.1: Overview of widely used parallel sparse linear solvers for large-scale scientific computations [13].

In the context of this work, the software QUANSCIENT ALLSOLVE relies on MUMPS as a direct solver and on PETSC for iterative strategies. While these solvers provide robust tools for handling large linear systems, their efficiency strongly depends on how the underlying problem is structured and distributed across the computational resources. This naturally leads to the consideration of domain decomposition techniques, which play a key role in enabling scalable and efficient parallel computations.

2.5.2 Domain Decomposition Methods

DDM form a distinct family of numerical strategies designed to solve large-scale systems by dividing the computational domain into smaller, more manageable subregions. Each subdomain is treated independently, allowing the use of local solvers, while an iterative procedure ensures consistency and convergence across the global interface. This hierarchical structure makes such approaches particularly suitable for parallel architectures, where multiple processors can operate simultaneously on different portions of the problem. Comprehensive discussions on both the mathematical framework and the algorithmic implementation of these methods are provided in [58]. A representative illustration of the mesh partitioning concept is presented in Figure 2.5.1.

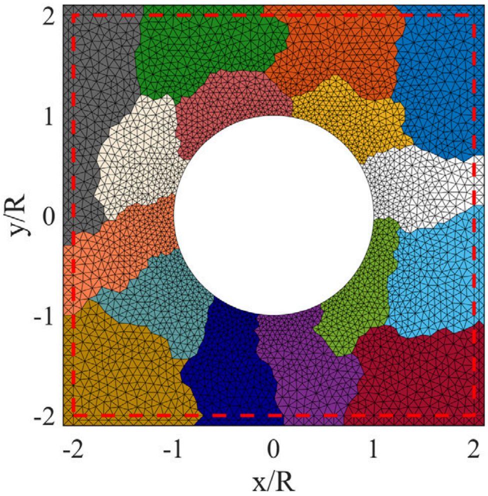


Figure 2.5.1: Example of partitioning of the discretized domain with sixteen sub-domains [59].

2.5.3 Parallelisation of the Harmonic Balance Method

Parallel implementations of the HBM have emerged across several scientific fields to address the increased computational cost associated with high-dimensional nonlinear systems. In aerodynamics, Jackson et al. [60] demonstrated that the enlarged memory footprint of multi-harmonic formulations motivates hybrid distributed shared memory strategies for harmonic balance Navier-Stokes solvers. In circuit analysis Dong and Li [61] explored coarse-grained parallelisation of nonlinear device evaluation and preconditioning operations, while Wang et al. [62] showed that GPU acceleration can significantly reduce the cost of assembling the nonlinear Jacobian matrix.

Recent developments have increasingly focused on parallel HBM formulations tailored to large-scale mechanical systems, where nonlinearities and mesh sizes can be substantial. Blahoš [63] proposed a scalable frequency-domain framework based on a predictor–corrector continuation method coupled with Newton–Raphson iterations. His study compared three parallel strategies for solving the linearised system at each Newton step, including a MUMPS direct solver, a PETSc GMRES iterative solver, and an in-house FETI implementation, highlighting the effectiveness of domain decomposition for large finite-element models.

Building on this foundation, Saponaro et al. [64] extended the FETI-based approach to the nonlinear forced response of bladed disks with friction contacts. Their work demonstrated that parallel HBM can accurately capture both geometric and contact-induced nonlinearities in realistic turbomachinery assemblies, confirming the suitability of FETI as a scalable solver strategy for industry-scale structural dynamics.

The present master's thesis follows directly in this line of research, extending the framework introduced by Blahoš [13] and further investigating the parallel solution of large-scale mechanical harmonic balance problems using domain decomposition approaches.

2.6 Conclusion

The reviewed literature covers the key concepts and methodologies involved in the analysis of nonlinear mechanical vibrations. The different types of nonlinearities were examined to clarify their physical origins, followed by a discussion of nonlinear analysis frameworks addressing both forced responses through NLFRs and intrinsic dynamics via NNMs. Frequency-domain techniques based on the HBM and numerical continuation were also reviewed, together with the linear and nonlinear solvers that support these approaches, including parallelisation strategies and DDM.

However, the parallelisation of the HBM for nonlinear mechanical systems remains comparatively limited in the existing body of work. While several contributions underline its potential, they often target restricted problem classes or exhibit limited scalability. This gap in the literature provides a clear motivation for the present thesis, which aims to further develop, implement, and evaluate parallel HBM strategies capable of addressing large-scale nonlinear vibration problems more effectively.

3

METHODOLOGY

This chapter presents the methodology developed for the nonlinear vibration analysis of the studied structures. The framework follows a rigorous and systematic progression, from the physical modelling of the problem to the implementation of numerical strategies for computing NLFRs and NNMs.

The formulation begins with the physical description in Section 3.1, where deformation measures, constitutive laws, and boundary conditions are introduced. This continuous description is then translated into a Finite Element Method (FEM) discretisation in Section 3.1.3, yielding the algebraic equations of motion together with the adopted damping models. The formulation is subsequently extended to account for steady rotation effects in Section 3.1.4. To investigate periodic behaviours, the resulting equations are reformulated in the frequency domain using the HBM, presented in Section 3.2, enabling the computation of steady-state periodic solutions in both forced and conservative cases.

The solution of the resulting nonlinear algebraic system is discussed in Section 3.3, where a Newton–Raphson scheme combined with bordered formulations is employed. Particular attention is given to the efficient evaluation of nonlinear terms, which can be achieved through the AFT method detailed in Section 3.2.1.

Building on this foundation, continuation techniques are introduced in Section 3.4 to trace solution branches efficiently and robustly. The framework is then extended to the computation of NNMs in Section 3.5, where additional ingredients such as a phase condition and a relaxation parameter are incorporated to ensure uniqueness and numerical stability.

Finally, the computational tools employed in this work are described. PYHARM (Section 3.7) was used as an academic validation environment for the developed methodology, whereas QUANSCIENT ALLSOLVE (Section 3.6) was leveraged for large-scale simulations on cloud-based high-performance computing infrastructures.

3.1 Problem modelling

Since the present approach relies on a FEM formulation, the underlying modelling framework follows a well-established and classical methodology in structural dynamics. Nevertheless, in order to ensure completeness and consistency with the subsequent developments, the physical formulation introduced in this section is largely based on the PhD thesis of Jiří Blahoš [13], from which the main kinematic and constitutive assumptions are recalled.

The analysis of structural vibrations is based on a continuum mechanics description, in which the geometry, material behaviour, external loads, and constraints are consistently represented through a displacement field defined over the reference configuration. From this field, strain measures and constitutive relations are introduced to characterise the mechanical response of the structure.

A linear elastic formulation is recovered under the assumption of small strains and rotations. When moderate or large deformations are involved, geometrical nonlinearities must be taken into account in order to capture the structural response accurately. Both linear and geometrically nonlinear kinematic descriptions are therefore considered in the present work, as detailed in Section 3.1.1.

The continuous problem is completed by the specification of boundary conditions, discussed in Section 3.1.2, and subsequently discretised using the finite element method, as described in Section 3.1.3. The formulation is finally extended to structures undergoing steady rotation by introducing a rotating reference frame to account for centrifugal effects, as presented in Section 3.1.4 is developed specifically within the present study and does not originate from the work of Blahoš.

3.1.1 Deformation

Consider an elastic body $\mathcal{B}_0 \subset \mathbb{R}^3$ in its reference configuration, described by a coordinate system $\mathbf{x}^0 \in \mathbb{R}^3$. When the body undergoes deformation, each point within its volume experiences a displacement relative to this reference configuration. As a result, the body occupies a new domain $\mathcal{B}$, characterised by a new coordinate system $\mathbf{x} : \mathbb{R}^3 \to \mathbb{R}^3$. The displacement field $\mathbf{u}$, which quantifies the deviation from the reference state, is given by:

$$\mathbf{u}(\mathbf{x}^0, t) = \mathbf{x}(\mathbf{x}^0, t) - \mathbf{x}^0, \quad \mathbf{u} : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^3. \tag{3.1.1}$$

The strain tensor can be expressed using the nabla operator as:

$$\boldsymbol{E} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T + (\nabla\mathbf{u})^T\nabla\mathbf{u}\right). \tag{3.1.2}$$

This expression can be simplified under the geometric linearity assumption, which consists of two conditions [65]:

- The extensional strains remain infinitesimal. Here, $\mathrm{tr}(\cdot)$ denotes the trace operator, i.e. the sum of the diagonal components of a tensor:

$$\mathrm{tr}(\nabla\mathbf{u}) = \nabla \cdot \mathbf{u} \ll 1. \tag{3.1.3}$$

- The rotations have small amplitudes. The operator $\mathrm{skew}(\cdot)$ extracts the skew-symmetric part of a tensor, which represents the local rigid-body rotation:

$$\|\mathrm{skew}(\nabla\mathbf{u})\| \ll 1. \tag{3.1.4}$$

Under these assumptions, the strain tensor simplifies to the linearised form:

$$\boldsymbol{\varepsilon} \approx \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{3.1.5}$$

In classical linear elasticity, the Cauchy stress tensor $\boldsymbol{\sigma}$ is related to the infinitesimal strain tensor ε through the constitutive law:

$$\boldsymbol{\sigma} = \boldsymbol{C}\,\varepsilon, \tag{3.1.6}$$

where $\boldsymbol{C}$ is the elasticity tensor, represented in matrix form with constant coefficients throughout the analysis. When a body undergoes large deformations, the assumption of small displacement gradients is no longer valid, and the complete expression of the Green–Lagrange strain tensor must be employed. In this framework, the stress measure commonly adopted is the second Piola–Kirchhoff stress tensor $\boldsymbol{S}$, which is entirely defined in the reference (undeformed) configuration and is work-conjugate to the Green–Lagrange strain tensor $\boldsymbol{E}$. For a Saint Venant–Kirchhoff material, small strains (though not necessarily small deformations) are assumed, leading to the constitutive relation:

$$\boldsymbol{S} = \boldsymbol{C}\,\boldsymbol{E}, \tag{3.1.7}$$

The balance of linear momentum in the reference configuration is then expressed as:

$$\nabla \cdot \boldsymbol{S} + \rho_0\,\boldsymbol{f_0} = \rho_0\,\ddot{\boldsymbol{u}}, \tag{3.1.8}$$

where $\boldsymbol{f_0}(\boldsymbol{x_0}, t)$ represents the external body force per unit reference volume, ρ_0 is the constant material density, and $\boldsymbol{u}$ is the displacement field.

3.1.2 Boundary Conditions

To define a well-posed boundary value problem, appropriate boundary conditions must be prescribed on the boundary $\partial\mathcal{B}_0$. The most common types are the Dirichlet, Neumann, Cauchy, and Robin conditions [13], which are applied on distinct portions of the boundary denoted by Γ_D, Γ_N, Γ_C, and Γ_R, respectively. The functions $\boldsymbol{a}(\boldsymbol{x}^0)$, $\boldsymbol{b}(\boldsymbol{x}^0)$, and $\boldsymbol{c}(\boldsymbol{x}^0)$ represent prescribed vector fields defined on $\partial\mathcal{B}_0$, depending on the boundary coordinate $\boldsymbol{x}^0$. In practical FEM workflows, these boundary subsets are typically identified through physical regions defined by the user during the mesh generation process, which group specific faces or edges of the geometry under a common label for the assignment of boundary conditions.

The Dirichlet boundary condition prescribes the displacement directly:

$$\boldsymbol{u} = \boldsymbol{a}(\boldsymbol{x}^0) \quad \text{on}\ \Gamma_D \subset \partial\mathcal{B}_0. \tag{3.1.9}$$

The Neumann boundary condition specifies the normal derivative of the field. In elasticity, this is commonly written as a prescribed traction vector:

$$\boldsymbol{\Sigma}\,\boldsymbol{n} = \boldsymbol{b}(\boldsymbol{x}^0) \quad \text{on}\ \Gamma_N \subset \partial\mathcal{B}_0, \tag{3.1.10}$$

where $\boldsymbol{\Sigma}$ is the stress tensor and $\boldsymbol{n}$ the outward normal to the boundary. Since the traction depends on gradients of $\boldsymbol{u}$, this condition is still a derivative (Neumann-type) constraint, as made explicit in Equation 3.1.2 and Equation 3.1.7.

The Cauchy condition imposes both the displacement and its normal derivative on the same portion of the boundary:

$$\left.\begin{aligned} \boldsymbol{u} &= \boldsymbol{a}(\boldsymbol{x}^0), \\ \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} &= (\nabla \boldsymbol{u})\boldsymbol{n} = \boldsymbol{b}(\boldsymbol{x}^0) \end{aligned}\right\} \quad \text{on } \Gamma_C \subset \partial\mathcal{B}_0. \tag{3.1.11}$$

Such conditions arise naturally in hyperbolic problems.

The Robin boundary condition is a weighted combination of Dirichlet and Neumann constraints[1]:

$$\alpha\, \boldsymbol{u} + \beta\, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = \boldsymbol{c}(\boldsymbol{x}^0) \quad \text{on } \Gamma_R \subset \partial\mathcal{B}_0. \tag{3.1.12}$$

In this work, only homogeneous Dirichlet boundary conditions are enforced, i.e.

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_D, \tag{3.1.13}$$

which corresponds to perfectly clamped regions of the structure.

3.1.3 Finite Elements

This section recalls the fundamentals of the FEM, with a particular focus on the weak formulation of the governing equations, which must be provided to QUANSCIENT for the finite element discretisation. A distinction is introduced between continuous and discrete quantities: the continuous displacement field is denoted by U, whereas $\boldsymbol{u}$ refers to the vector of nodal degrees of freedom. The reference configuration $\mathcal{B}_0$ is partitioned into non-overlapping elements e, such that

$$\mathcal{B}_0 = \bigcup_e e. \tag{3.1.14}$$

Each element is characterised by its topological entities nodes, edges, faces, and interiors. For each entity and each spatial direction x, y, z, a test function v is defined. These functions form a hierarchical basis, where each basis function is associated with a geometric entity and polynomial order. The restriction of a basis function to an element yields its shape function, which has compact support.

Let v_i denote the test functions indexed by $i \in \{1, \dots, N_{\text{dof}}\}$. The continuous displacement field is approximated as

$$U(x,t) = \sum_{i=1}^{N_{\text{dof}}} \boldsymbol{u}_i(t)\, v_i(x). \tag{3.1.15}$$

Starting from the strong form of the nonlinear elasticity Equation 3.1.8, the weak form is

---

[1] Cauchy boundary conditions resemble initial conditions for second-order ODEs, as they impose both position and slope at a single boundary. Dirichlet and Neumann conditions, by contrast, parallel specifying conditions at opposite ends of an interval.

obtained by multiplying by a test function and integrating over $\mathcal{B}_0$:

$$\rho_0 \int_{\mathcal{B}_0} \ddot{U}\, \boldsymbol{v}\, d\mathcal{B}_0 + \int_{\mathcal{B}_0} \boldsymbol{S} : \delta \boldsymbol{E}\, d\mathcal{B}_0 = \rho_0 \int_{\mathcal{B}_0} f_0\, \boldsymbol{v}\, d\mathcal{B}_0, \tag{3.1.16}$$

where $\delta\boldsymbol{E}$ denotes the variation of the Green–Lagrange strain tensor with respect to the test function v.

Each component of Equation 3.1.16 is interpreted as follows:

- **Inertia term:**

$$\boldsymbol{M}\, \ddot{\boldsymbol{u}} = \rho_0 \int_{\mathcal{B}_0} \ddot{U}\, \boldsymbol{v}\, d\mathcal{B}_0. \tag{3.1.17}$$

- **External force:**

$$\boldsymbol{f}_{ext} = \rho_0 \int_{\mathcal{B}_0} f_0\, \boldsymbol{v}\, d\mathcal{B}_0. \tag{3.1.18}$$

- **Stiffness and nonlinear internal force term:** Following the derivation detailed by Blahoš [13], the expansion of the strain energy with respect to $\boldsymbol{u}$ yields the classical linear stiffness contribution together with a nonlinear internal force term,

$$\boldsymbol{K}\, \boldsymbol{u} + \boldsymbol{f}_{nl}(\boldsymbol{u}) = \boldsymbol{f}_{nl}^{int} = \int_{\mathcal{B}_0} \boldsymbol{S} : \delta \boldsymbol{E}\, d\mathcal{B}_0. \tag{3.1.19}$$

  The nonlinear contribution $\boldsymbol{f}_{nl}$ arises from the geometric terms in the Green-Lagrange strain tensor, whose expansion generates quadratic and cubic components [6, 66, 67]. For symmetric configurations such as clamped–clamped beams, the quadratic contribution vanishes identically, so that the geometric nonlinearity is dominated by cubic terms. These effects are automatically accounted for by the `predefinedelasticity` routine in Quanscient Allsolve, which constructs the nonlinear internal force vector $\boldsymbol{f}_{nl}^{int}$.

The complete discretised weak form thus yields the nonlinear structural dynamics system:

$$\boldsymbol{M}\, \ddot{\boldsymbol{u}} + \boldsymbol{D}\, \dot{\boldsymbol{u}} + \boldsymbol{K}\, \boldsymbol{u} + \boldsymbol{f}_{nl}(\boldsymbol{u}) = \boldsymbol{f}_{ext}(t). \tag{3.1.20}$$

Damping is introduced via Rayleigh damping,

$$\boldsymbol{D} = \alpha\, \boldsymbol{M} + \beta\, \boldsymbol{K}, \tag{3.1.21}$$

where α and β are the Rayleigh damping coefficients. Although this formulation is widely used due to its simplicity and because it preserves modal orthogonality in the linear case, it lacks a direct physical basis and can produce non-physical damping forces when applied to nonlinear systems. In particular, relying on the initial stiffness $\boldsymbol{K}$ may lead to overdamping in post-yield or softening regimes, thereby distorting the system's energy dissipation and dynamic response [68, 69]. To alleviate these drawbacks, alternatives such as the Maxwell damping model have been proposed, in which the stiffness-proportional dashpot is replaced by one or more Maxwell elements, extending the frequency range of nearly constant damping and reducing spurious effects [70].

The final form of the equation of motion, including the damping term, is given by:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{D}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} + \boldsymbol{f}_{nl}(\boldsymbol{u}) = \boldsymbol{f}_{ext}(\omega, t). \tag{3.1.22}$$

These equations will be studied in the context of vibrations, which typically lead to periodic solutions. In applications where part of the structure rotates, the formulation must be extended to a rotating reference frame to incorporate rotation-induced effects.

3.1.4 Rotating Frame

When the structure experiences steady rotation, additional terms arise in the equation of motion, most notably the centrifugal effect and the Coriolis contribution. The latter can often be neglected, particularly when the rotation axis is orthogonal to the dominant vibration directions, as shown in [67]. A complete derivation of the rotating-frame formulation and its associated matrices is provided in the PhD thesis of Salles [18]. Under this assumption, the equation of motion becomes:

$$\boldsymbol{M}\ddot{\boldsymbol{p}} + \boldsymbol{D}\dot{\boldsymbol{p}} + (\boldsymbol{K} + \boldsymbol{K}_{\Omega})\boldsymbol{p} + \boldsymbol{f}_{nl}(\boldsymbol{p}) = \boldsymbol{f}_{ext}(t) + \boldsymbol{f}_{\Omega}(\varpi)\,. \tag{3.1.23}$$

Where p is refere to the equations of motion write for the unknown displacement $\boldsymbol{p}(t, \varpi)$ Two additional terms are introduced by the presence of rotation:

- Centrifugal softening stiffness matrix:

$$\boldsymbol{K}_{\Omega} = \int_{\mathcal{B}_0} \rho\, \boldsymbol{v}^T \boldsymbol{\Omega}^2 \boldsymbol{v}\; d\mathcal{B}_0\,. \tag{3.1.24}$$

- Centrifugal load:

$$\boldsymbol{f}_{\Omega} = \int_{\mathcal{B}_0} \rho\, \boldsymbol{v}^T \boldsymbol{\Omega}^2 \boldsymbol{r}\; d\mathcal{B}_0\,. \tag{3.1.25}$$

In these expressions, $\boldsymbol{\Omega}$ denotes the skew-symmetric second-order rotation tensor, and $\boldsymbol{r}$ represents the position vector of an arbitrary point of the structure expressed in the rotating frame.

The rotation speed is assumed to be constant. Under this assumption, the centrifugal loading produces a static deformation of the structure, leading to a new equilibrium configuration. The dynamic behaviour is therefore characterised by vibrations about this centrifugally prestressed state. The total displacement field is decomposed as

$$\boldsymbol{p}(t, \varpi) = \boldsymbol{u}_0(\varpi) + \boldsymbol{u}(t), \tag{3.1.26}$$

where $\boldsymbol{u}_0(\varpi)$ denotes the static equilibrium displacement induced by steady rotation, and $\boldsymbol{u}(t)$ represents the time-dependent vibration about this equilibrium configuration. The deformed configuration associated with a constant rotation speed is obtained by solving the nonlinear static equilibrium problem

$$(\boldsymbol{K} + \boldsymbol{K}_{\Omega})\boldsymbol{u}_0 + \boldsymbol{f}_{nl}(\boldsymbol{u}_0) = \boldsymbol{f}_{\Omega}(\varpi) + \boldsymbol{f}_{ext}. \tag{3.1.27}$$

By introducing the decomposition Equation 3.1.26 into the full equations of motion Equation 3.1.23, and by making use of the static equilibrium condition Equation 3.1.27, the governing equations for the nonlinear vibrations about the centrifugally deformed configuration are obtained as

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{D}\dot{\boldsymbol{u}} + \boldsymbol{K}_\Omega \boldsymbol{u} + \left(\boldsymbol{f}_{nl}^{int}(\boldsymbol{u}_0 + \boldsymbol{u}) - \boldsymbol{f}_{nl}^{int}(\boldsymbol{u}_0)\right) = \boldsymbol{f}_{ext}(t). \tag{3.1.28}$$

This formulation provides an exact description of the nonlinear dynamic response of the structure in the vicinity of the centrifugally prestressed equilibrium configuration.

The implementation is verified in Section 5.4, where the test case is validated against an analytical representation of the static behaviour and further compared with a nonlinear Campbell diagram and NNMs constructed using the same properties as those reported in [67].

3.1.4.1 Campbell diagram

Campbell diagrams are widely used in rotordynamics and turbomachinery [7, 18, 67] to analyse the evolution of natural frequencies with rotation speed and to identify potential resonances with rotation-induced excitation frequencies

For each rotation speed, the nonlinear static equilibrium problem Equation 3.1.27 is first solved to determine the prestressed configuration $\boldsymbol{u}_0(\varpi)$. Small-amplitude vibrations are then considered as perturbations about this equilibrium state. The nonlinear internal force vector is linearised with respect to the displacement around $\boldsymbol{u}_0$, leading to the first-order expansion

$$\boldsymbol{f}_{nl}^{int}(\boldsymbol{u}_0 + \boldsymbol{u}) \approx \boldsymbol{f}_{nl}^{int}(\boldsymbol{u}_0) + \boldsymbol{K}_t(\boldsymbol{u}_0)\boldsymbol{u}, \tag{3.1.29}$$

where the tangent stiffness matrix evaluated at the static equilibrium configuration is defined as

$$\boldsymbol{K}_t(\boldsymbol{u}_0) = \left.\frac{\partial \boldsymbol{f}_{nl}^{int}}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}=\boldsymbol{u}_0}. \tag{3.1.30}$$

Introducing the linearised internal force into the governing equation Equation 3.1.28 yields the linearised equation describing small vibrations about the centrifugally prestressed state,

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{D}\dot{\boldsymbol{u}} + \left(\boldsymbol{K}_\Omega + \boldsymbol{K}_t(\boldsymbol{u}_0)\right)\boldsymbol{u} = \boldsymbol{f}_{\text{ext}}(t). \tag{3.1.31}$$

In the absence of external excitation, the natural frequencies of the prestressed structure are obtained by solving the associated eigenvalue problem

$$\left(\boldsymbol{K}_\Omega + \boldsymbol{K}_t(\boldsymbol{u}_0) - \omega^2 \boldsymbol{M}\right)\boldsymbol{\phi} = \boldsymbol{0}. \tag{3.1.32}$$

Repeating this procedure for increasing rotation speeds allows the evolution of the natural frequencies with respect to Ω to be tracked, yielding the Campbell diagram. From this representation, critical speeds, frequency veering phenomena, and mode interactions can be identified [19]. Since the eigenvalue problems associated with different rotation speeds are mutually independent, the solution at a given rotational speed does not influence the frequencies obtained at other speeds. As a result, these computations can be efficiently parallelised using independent threads. In this context, the `allsolve` function provided by Quanscient can be employed directly to compute the full set of solutions in parallel.

3.2 Harmonic Balance Method

Vibrations can be classified into two types [13]:

- **Free vibration – Normal modes**: This type of motion occurs in the absence of external forces. When damping is neglected, the response remains purely periodic, with amplitudes governed solely by the system's physical properties. In the present work, damping is neglected, and the motion therefore remains strictly periodic.
- **Forced vibration – Frequency response**: This type of motion arises when the system is subjected to an external periodic excitation. In steady-state conditions, the response becomes periodic, while any transient components vanish.

The problems considered in this work are governed by the equation of motion previously established in Equation 3.1.22. The same equation may also be formulated in a rotating reference frame; however, this extension is not developed here[2]. A natural way to describe periodic responses consists in expressing the solution using sine and cosine functions, leading to the concept of harmonic motion. This forms the basis of the HBM, which assumes the system response to be periodic and representable by a finite number of harmonics. The external excitation force $\boldsymbol{f}_{\text{ext}}(t)$ is assumed to be sinusoidal.

In nonlinear systems [11], an excitation at angular frequency ω may generate response components not only at this frequency, but also at its integer multiples or fractions, due to the presence of nonlinear internal forces. Consequently, the response must be expanded over several harmonics. The following derivation follows the organisation and formalism proposed by Detroux *et al.* [11], within the standard HBM framework.

The displacement vector $\boldsymbol{u}(t)$ is approximated by a truncated Fourier series,

$$\boldsymbol{u}(t) = \frac{\tilde{\boldsymbol{u}}_0}{\sqrt{2}} + \sum_{k=1}^{n_H} \left(\tilde{\boldsymbol{u}}_k^c \cos\left(\frac{k\omega t}{\nu}\right) + \tilde{\boldsymbol{u}}_k^s \sin\left(\frac{k\omega t}{\nu}\right) \right), \tag{3.2.1}$$

where $\tilde{\boldsymbol{u}}_0$ denotes the constant (zero-frequency) component of the response, while $\tilde{\boldsymbol{u}}_k^c$ and $\tilde{\boldsymbol{u}}_k^s$ are the cosine and sine coefficient vectors associated with the k-th harmonic, respectively. These coefficients are time-independent and constitute the unknowns of the harmonic balance problem. The parameter n_H defines the number of retained harmonics, while ν accounts for the possible presence of sub-harmonic [24] responses with respect to the excitation frequency ω.

This formulation corresponds to a finite Fourier series approximation of the periodic solution. Since the number of harmonics is finite, the representation is approximate, and increasing n_H generally improves the accuracy of the solution [13]. The nonlinear internal force $\boldsymbol{f}_{\text{nl}}(\boldsymbol{u})$, the external excitation force $\boldsymbol{f}_{\text{ext}}(\omega, t)$ are combined into a single force vector,

$$\boldsymbol{f}(\boldsymbol{u}, \omega, t) = \boldsymbol{f}_{\text{ext}}(\omega, t) - \boldsymbol{f}_{\text{nl}}(\boldsymbol{u}), \tag{3.2.2}$$

---

[2] In a rotating-frame formulation, the centrifugal stiffness contribution may be grouped with the structural stiffness matrix, $(\boldsymbol{K} + \boldsymbol{K}_\Omega)$, while the centrifugal load is treated as an external force term $\boldsymbol{f}_\Omega$.

which is approximated using the same harmonic basis as the displacement field,

$$\boldsymbol{f}(t) = \frac{\tilde{\boldsymbol{f}}_0}{\sqrt{2}} + \sum_{k=1}^{n_H} \left(\tilde{\boldsymbol{f}}_k^c \cos\left(\frac{k\omega t}{\nu}\right) + \tilde{\boldsymbol{f}}_k^s \sin\left(\frac{k\omega t}{\nu}\right) \right). \tag{3.2.3}$$

The Fourier coefficients $\tilde{\boldsymbol{f}}_k^c$ and $\tilde{\boldsymbol{f}}_k^s$ depend implicitly on the displacement coefficients $\tilde{\boldsymbol{u}}_k^c$ and $\tilde{\boldsymbol{u}}_k^s$, which therefore constitute the unknowns of the problem. These coefficients are gathered into vectors of size $(2n_H + 1)n \times 1$, where n denotes the number of degrees of freedom of the system,

$$\tilde{\boldsymbol{u}} = \left[(\tilde{\boldsymbol{u}}_0)^T \quad (\tilde{\boldsymbol{u}}_1^s)^T \quad (\tilde{\boldsymbol{u}}_1^c)^T \quad \dots \quad (\tilde{\boldsymbol{u}}_{n_H}^s)^T \quad (\tilde{\boldsymbol{u}}_{n_H}^c)^T \right]^T, \tag{3.2.4}$$

$$\tilde{\boldsymbol{f}} = \left[(\tilde{\boldsymbol{f}}_0)^T \quad (\tilde{\boldsymbol{f}}_1^s)^T \quad (\tilde{\boldsymbol{f}}_1^c)^T \quad \dots \quad (\tilde{\boldsymbol{f}}_{n_H}^s)^T \quad (\tilde{\boldsymbol{f}}_{n_H}^c)^T \right]^T. \tag{3.2.5}$$

The displacement and force expressions are then reformulated in a more compact form [11]:

$$\mathbf{u}(t) = (\mathbf{Q}(t) \otimes \mathbb{I}_n)\tilde{\boldsymbol{u}}, \quad \mathbf{f}(t) = (\mathbf{Q}(t) \otimes \mathbb{I}_n)\tilde{\boldsymbol{f}}, \tag{3.2.6}$$

where $\otimes$ denotes the Kronecker tensor product, $\mathbb{I}_n$ is the identity matrix of size n, and $\mathbf{Q}(t)$ is a vector containing the sine and cosine series:

$$\mathbf{Q}(t) = \left[\frac{1}{\sqrt{2}} \quad \sin\left(\frac{\omega t}{\nu}\right) \quad \cos\left(\frac{\omega t}{\nu}\right) \quad \dots \quad \sin\left(n_H \frac{\omega t}{\nu}\right) \quad \cos\left(n_H \frac{\omega t}{\nu}\right) \right]. \tag{3.2.7}$$

Velocities and accelerations can also be expressed using the Fourier series [71], leading to

$$\dot{\mathbf{u}}(t) = (\dot{\mathbf{Q}}(t) \otimes \mathbb{I}_n)\tilde{\boldsymbol{u}} = ((\mathbf{Q}(t)\tilde{\nabla}) \otimes \mathbb{I}_n)\tilde{\boldsymbol{u}}, \tag{3.2.8}$$

$$\ddot{\mathbf{u}}(t) = (\ddot{\mathbf{Q}}(t) \otimes \mathbb{I}_n)\tilde{\boldsymbol{u}} = \left((\mathbf{Q}(t)\tilde{\nabla}^2) \otimes \mathbb{I}_n\right)\tilde{\boldsymbol{u}}. \tag{3.2.9}$$

where the differentiation operator in the frequency domain is given by :

$$\tilde{\nabla} = \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \tilde{\nabla}_k & & \\ & & & \ddots & \\ & & & & \tilde{\nabla}_{n_H} \end{bmatrix}, \qquad \tilde{\nabla}\tilde{\nabla} = \tilde{\nabla}^2 = \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \tilde{\nabla}_k^2 & & \\ & & & \ddots & \\ & & & & \tilde{\nabla}_{n_H}^2 \end{bmatrix}. \tag{3.2.10}$$

Each block $\tilde{\nabla}_k$ and its square $\tilde{\nabla}_k^2$ are given by

$$\tilde{\nabla}_k = \begin{bmatrix} 0 & -\frac{k\omega}{\nu} \\ \frac{k\omega}{\nu} & 0 \end{bmatrix}, \qquad \tilde{\nabla}_k^2 = \begin{bmatrix} -\left(\frac{k\omega}{\nu}\right)^2 & 0 \\ 0 & -\left(\frac{k\omega}{\nu}\right)^2 \end{bmatrix}. \tag{3.2.11}$$

Substituting Equations 3.2.6, 3.2.8, and 3.2.9 into the equations of motion Equation 3.1.22

yields

$$\left(\left(\mathbf{Q}(t)\tilde{\nabla}^2\right)\otimes\mathbf{M}\right)\tilde{\boldsymbol{u}}+\left(\left(\mathbf{Q}(t)\tilde{\nabla}\right)\otimes\mathbf{D}\right)\tilde{\boldsymbol{u}}+\left(\left(\mathbf{Q}(t)\otimes\mathbf{K}\right)\right)\tilde{\boldsymbol{u}}=\left(\mathbf{Q}(t)\otimes\mathbb{I}_n\right)\tilde{\boldsymbol{f}}. \tag{3.2.12}$$

To eliminate the time dependency and obtain an expression relating the different Fourier coefficients, a Galerkin procedure is applied by projecting the equation onto the orthogonal trigonometric basis $\boldsymbol{Q}(t)$. This requires the projection of the equation onto the same harmonic functions used to approximate the solution:

$$\begin{aligned}&\left(\left(\frac{2}{T}\int_0^T\mathbf{Q}^T(t)\mathbf{Q}(t)\,dt\,\tilde{\nabla}^2\right)\otimes\mathbf{M}\right)\tilde{\boldsymbol{u}}+\left(\left(\frac{2}{T}\int_0^T\mathbf{Q}^T(t)\mathbf{Q}(t)\,dt\,\tilde{\nabla}\right)\otimes\mathbf{D}\right)\boldsymbol{u}\\&+\left(\left(\frac{2}{T}\int_0^T\mathbf{Q}^T(t)\mathbf{Q}(t)\,dt\right)\otimes\mathbf{K}\right)\tilde{\boldsymbol{u}}=\left(\left(\frac{2}{T}\int_0^T\mathbf{Q}^T(t)\mathbf{Q}(t)\,dt\right)\otimes\mathbb{I}_n\right)\tilde{\boldsymbol{f}},\end{aligned} \tag{3.2.13}$$

where T is the period of the external force. Considering that

$$\frac{2}{T}\int_0^T\mathbf{Q}^T(t)\mathbf{Q}(t)dt=\mathbb{I}_{2n_H+1}, \tag{3.2.14}$$

the equations of motion expressed in the frequency domain are obtained as

$$\left(\tilde{\nabla}^2\otimes\mathbf{M}\right)\tilde{\boldsymbol{u}}+\left(\tilde{\nabla}\otimes\mathbf{D}\right)\tilde{\boldsymbol{u}}+\left(\mathbb{I}_{2n_H+1}\otimes\mathbf{K}\right)\tilde{\boldsymbol{u}}=\left(\mathbb{I}_{2n_H+1}\otimes\mathbb{I}_n\right)\tilde{\boldsymbol{f}}, \tag{3.2.15}$$

or, in a more compact form,

$$\mathbf{G}(\tilde{\boldsymbol{u}},\omega)\equiv\mathbf{Z}(\omega)\tilde{\boldsymbol{u}}-\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{u}})=0. \tag{3.2.16}$$

Where $\mathbf{Z}$ is the $(2n_H+1)n\times(2n_H+1)n$ matrix describing the linear dynamics:

$$\begin{aligned}\mathbf{Z}&=\tilde{\nabla}^2\otimes\mathbf{M}+\tilde{\nabla}\otimes\mathbf{D}+\mathbb{I}_{2n_H+1}\otimes\mathbf{K},\\&=\begin{bmatrix}\mathbf{K}&&&&&\\&\mathbf{K}-\left(\frac{\omega}{\nu}\right)^2\mathbf{M}&-\frac{\omega}{\nu}\mathbf{D}&&&\\&\frac{\omega}{\nu}\mathbf{D}&\mathbf{K}-\left(\frac{\omega}{\nu}\right)^2\mathbf{M}&&&\\&&&\ddots&&\\&&&&\mathbf{K}-\left(n_H\frac{\omega}{\nu}\right)^2\mathbf{M}&-n_H\frac{\omega}{\nu}\mathbf{D}\\&&&&n_H\frac{\omega}{\nu}\mathbf{D}&\mathbf{K}-\left(n_H\frac{\omega}{\nu}\right)^2\mathbf{M}\end{bmatrix}.\end{aligned} \tag{3.2.17}$$

The nonlinear force vector $\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{u}})$ can be separated into two contributions like the Equation 3.2.2: the external excitation $\tilde{\boldsymbol{f}}_{\text{ext}}$ and the nonlinear restoring force $\tilde{\boldsymbol{f}}_{\text{nl}}(\tilde{\boldsymbol{u}},\omega)$. The full residual equation becomes

$$\mathbf{G}(\tilde{\boldsymbol{u}},\omega)\equiv\mathbf{Z}(\omega)\tilde{\boldsymbol{u}}+\tilde{\boldsymbol{f}}_{\text{nl}}(\tilde{\boldsymbol{u}},\omega)-\tilde{\boldsymbol{f}}_{\text{ext}}=0. \tag{3.2.18}$$

In cases where an explicit expression relates the nonlinear force to the Fourier coefficients of the displacement, a direct formulation can be employed like the one with the geometrical nonlinearity. However, for certain types of non-linearities, such a relationship cannot be established. In these cases, the AFT method is required, as introduced in Section 3.2.1.

3.2.1 Alternating Frequency-Time Method

For the solution of nonlinear systems, the evaluation of the nonlinear force and its associated Jacobian is generally required, for instance when Newton-type solvers are employed, as introduced in the next section. When closed-form expressions for these quantities are not available AFT [72] method offers an efficient alternative. The principle of AFT is based on a double transformation between frequency and time domains. The harmonic displacement vector $\tilde{\boldsymbol{u}}$ is first mapped into the time domain using an Inverse Fast Fourier Transform (IFFT), allowing the nonlinear force to be evaluated pointwise in time. This time-domain response is then transformed back to the frequency domain via a Fast Fourier Transform (FFT) to obtain $\tilde{\boldsymbol{f}}$ [3]:

$$\tilde{\boldsymbol{u}} \xrightarrow{\text{IFFT}} \boldsymbol{u}(t) \longrightarrow \boldsymbol{f}_{nl}(\boldsymbol{u}) \xrightarrow{\text{FFT}} \tilde{\boldsymbol{f}}_{nl}(\tilde{\boldsymbol{u}}). \tag{3.2.19}$$

The Jacobian $\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}$ can be estimated by finite differences, though this approach is computationally intensive. A key numerical parameter introduced in this context is the number of time samples N, which defines the resolution of the discretised period T through $\Delta_t = \frac{T}{N}$. To avoid aliasing effects, the Shannon sampling theorem requires $N \geq 2n_H + 1$, where n_H denotes the number of harmonics.

In Quanscient Allsolve, the Fourier-based Jacobian is evaluated analytically whenever possible. When the AFT method is enabled, as in the present work, the number of time samples is systematically set to $N = 2(2n_H + 1)$ to ensure sufficient resolution and to avoid aliasing effects. A comparison between the analytical and AFT formulations is presented in Section 5.5.2.1.

3.3 Nonlinear Solver

The nonlinear residual equation introduced in Equation 3.2.16 cannot be solved analytically and must therefore be addressed using an iterative numerical procedure. Among the most established techniques, the Newton–Raphson method [73] remains the standard approach due to its rapid and quadratic convergence in the vicinity of the solution. When the initial estimate lies sufficiently close to the true equilibrium point, the method exhibits quadratic convergence [73]. It has been successfully employed in the context of nonlinear vibration analyses, as demonstrated in several studies [11, 13].

In practice, Newton's method may be applied either to the unknown vector $\tilde{\boldsymbol{u}}$ alone or to an augmented formulation that includes one or more continuation parameters, denoted collectively by $\boldsymbol{y}$. The inclusion of these parameters generally renders the system underdetermined, which necessitates the introduction of an additional constraint equation to ensure solvability. The resulting extended system is subsequently resolved through a bordering algorithm, as detailed in Section 3.3.1.

The iterative process begins with an initial estimate $\boldsymbol{y}^k$. At each iteration, a correction increment $\Delta\boldsymbol{y}^k$ is obtained by solving the linearised system:

$$\nabla\boldsymbol{G}(\boldsymbol{y}^k)\,\Delta\boldsymbol{y}^k = -\boldsymbol{G}(\boldsymbol{y}^k), \tag{3.3.1}$$

---

[3] A discrete Fourier transform (DFT) and its inverse (IDFT) may also be used to perform this operation.

where $\nabla \boldsymbol{G}$ denotes the Jacobian matrix of the residual vector $\boldsymbol{G}$. The solution is then updated according to:

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^{k} + \Delta \boldsymbol{y}^{k}. \tag{3.3.2}$$

The initial estimate $\boldsymbol{y}^0$ is typically provided by the predictor stage of the continuation algorithm, which ensures proximity to the expected branch of solutions, as discussed in Section 3.4. When no predictor is available, the process can start from a null vector, although convergence in that case is not guaranteed.

The convergence of Newton's iterations is governed by a set of stopping criteria. Commonly, these include the absolute residual norm $||\boldsymbol{G}(\boldsymbol{y}^k)||$, the relative residual $\frac{||\boldsymbol{G}(\boldsymbol{y}^k)||}{||\boldsymbol{G}(\boldsymbol{y}^0)||}$, or the stagnation of successive corrections quantified by $||\Delta \boldsymbol{y}^k||$. Each of these quantities must fall below a prescribed tolerance ϵ, while the total number of iterations is limited by a maximum value to prevent divergence. When properly tuned, these conditions ensure convergence towards the physically consistent solution branch.

The choice of tolerance ϵ is critical to achieving reliable convergence. An excessively large value may cause premature termination on an incorrect branch or in non-physical regions of the solution space, while an overly restrictive tolerance can prevent convergence altogether due to round-off accumulation, particularly in near-resonant configurations where the system is ill-conditioned. To address this limitation, QUANSCIENT ALLSOLVE provides a *quadruple precision* option based on the IEEE 754 `float128` standard, which significantly reduces floating-point sensitivity. The benefit of this enhanced precision is illustrated in Section 5.5.2.2, where results obtained in double and quadruple precision are compared. The variation of the tolerance parameter ϵ along a continuation path is shown in Figure 3.3.1, providing a direct visualisation of the convergence behaviour.

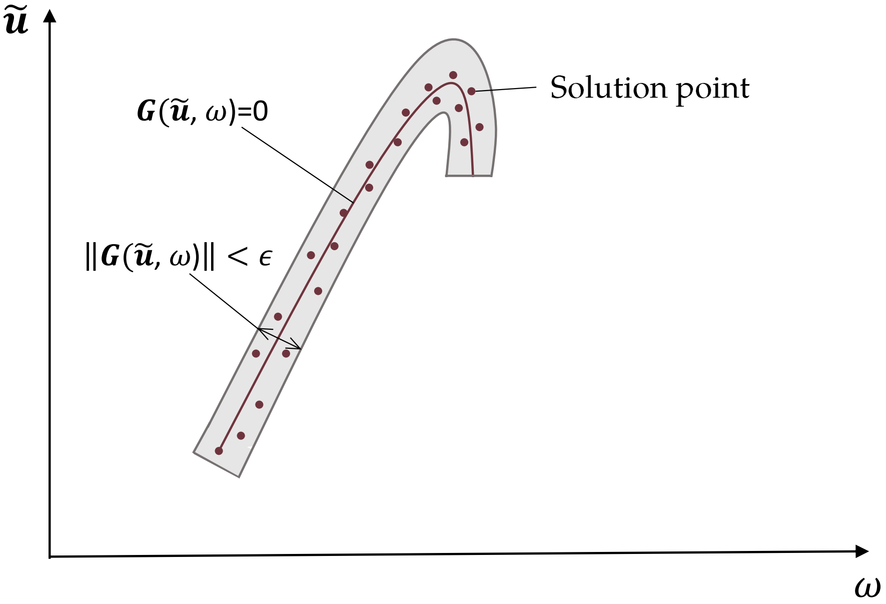


Figure 3.3.1: Evolution of the convergence tolerance ϵ along a continuation branch, adapted from [74].

An implementation of Newton's method with QUANSCIENT for fixed-frequency continuation is provided in Algorithm 1.

Algorithm 1 Newton's method at fixed ω.

```
Input: $\omega$, tol, $\max_{\text{iter}}$
SetFundamentalFrequency($\omega$)
$\tilde{\boldsymbol{u}}^k = \mathbf{0}$, iter = 0, relative_error = 1
while iter < $\max_{\text{iter}}$ do
    $\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k, \boldsymbol{b}^k$ = GenerateJac&b(elasticity)
    $\boldsymbol{G}^k = \boldsymbol{b}^k - \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k\tilde{\boldsymbol{u}}^k$
    if $\|\boldsymbol{G}^k\| <$ tol then
        break
    end if
    $\tilde{\boldsymbol{u}}^{k+1}$ = Solve($\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k, \boldsymbol{b}^k$)
    SetData($\tilde{\boldsymbol{u}}^{k+1}$)
    $\tilde{\boldsymbol{u}}^k = \tilde{\boldsymbol{u}}^{k+1}$
    iter ← iter + 1
end while
Return: $\tilde{\boldsymbol{u}}^k, \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k$
```

3.3.1 Bordering Algorithm

To solve large-scale systems containing a main matrix, here the Jacobian, bordering algorithms are employed [75]. In this thesis, they are used multiple times, particularly when the Newton–Raphson method is applied to multiple continuation parameters like the Equation 3.5.17, but also in other situations. Such systems can be expressed as

$$\begin{bmatrix} \mathbf{J} & \mathbf{A} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix}, \tag{3.3.3}$$

where $\mathbf{J} \in \mathbb{R}^{n\times n}$, $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, and $\mathbf{C} \in \mathbb{R}^{m\times m}$. Typically, $n \gg m$.

The bordering algorithm applies a Gauss–Legendre elimination around the rows and columns corresponding to the additional unknowns and constraints [13]. The procedure involves the following steps:

$$\begin{aligned} \mathbf{J}\mathbf{X}_1 &= \mathbf{F}, \\ \mathbf{J}\mathbf{X}_2 &= \mathbf{A}, \\ (\mathbf{C} - \mathbf{B}^T\mathbf{X}_2)\mathbf{Y} &= \mathbf{G} - \mathbf{B}^T\mathbf{X}_1, \\ \mathbf{X} &= \mathbf{X}_1 - \mathbf{X}_2\mathbf{Y}. \end{aligned} \tag{3.3.4}$$

The principal advantage of this approach is that it allows solving linear systems involving only $\mathbf{J}$, the Jacobian matrix associated with the regular solution point, which is assumed to be nonsingular [76]. As a result, the computationally expensive factorisation of $\mathbf{J}$ must be performed only once, which significantly improves efficiency compared to directly factorising the larger extended system. In practice, this reduction in computational cost proved substantial: when the simulation was performed without reusing the factorisation, the overall runtime increased by approximately 30 %.

3.4 Continuation

Continuation constitutes the central methodology of this thesis. It refers to the process of tracing the solutions of a nonlinear system as one of its parameters is varied, thereby generating a solution branch. In this work, attention is restricted to a single branch, and the governing equations are analysed in terms of the residual defined in Equation 3.2.16. Several continuation strategies are examined. First, the notion of a regular solution is introduced in Section 3.4.1. Then, the predictor–corrector approach is outlined (Section 3.4.2), beginning with the simplest case of natural parameter continuation presented in Section 3.4.3, followed by more advanced predictor–corrector schemes in Section 3.4.4. Step-size adaptation techniques are subsequently discussed in Section 3.4.5, before these components are combined into a global continuation loop in Section 3.4.6.

3.4.1 Regular Solution

A solution $\tilde{\boldsymbol{u}}_i, \omega_i$ of $\boldsymbol{G}(\tilde{\boldsymbol{u}}, \omega) = 0$ is considered regular [77] if

$$\operatorname{Rank}\big(\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G} \mid \nabla_{\omega}\boldsymbol{G}\big) = n, \tag{3.4.1}$$

where the Jacobian $\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G} \mid \nabla_{\omega}\boldsymbol{G}$ has dimensions $n \times (n+1)$. Satisfying this condition ensures that the solution is locally unique and that a unique one-dimensional continuum of solutions $(\tilde{\boldsymbol{u}}(s), \omega(s))$, known as a solution family or solution branch, passes through $(\tilde{\boldsymbol{u}}_i, \omega_i)$.

Once the existence of a unique local solution is ensured through the regularity condition, various algorithms can be applied to track this solution as parameters vary. Among these, predictor–corrector schemes form the backbone of most continuation methods.

3.4.2 Predictor–Corrector Schemes

Predictor–corrector schemes trace a curve of solutions by alternating between two steps. The predictor estimates the next point on the branch from previously computed points, while the corrector refines this estimate by solving the nonlinear system. The predictor step involves both the choice of algorithm and the step length used to advance along the branch, both of which influence the efficiency and robustness of the continuation process. The strategies adopted in this work for controlling the step length are presented in Section 3.4.5. The choice of predictor and corrector algorithms can be adapted depending on the characteristics of the branch being followed.

3.4.3 Natural Parameter Continuation

The natural continuation method, illustrated in Figure 3.4.1, employs a trivial predictor where the predicted location is given by the previous solution, $\tilde{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}^p_{i+1}$. The corrector stage applies Newton's method while keeping the frequency fixed, as implemented in Algorithm 1. This process can be summarised as

$$\begin{bmatrix} \boldsymbol{G}(\boldsymbol{y}) \\ \omega_{i+1} - (\omega_i + s) \end{bmatrix} = \boldsymbol{0}, \tag{3.4.2}$$

where s is the continuation step size.

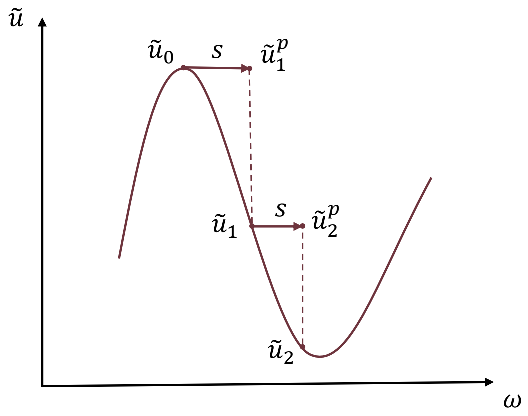


Figure 3.4.1: Natural parameter continuation, inspired by [78].

3.4.4 Advanced Predictors and Correctors

While natural parameter continuation offers a simple framework for branch following, it remains limited in its ability to handle folds or strong curvature in the solution path. To improve robustness and extend the range of convergence, more advanced predictors and correctors have been developed. In this work, the tangent and secant predictors are combined with pseudo-arclength and arclength correctors, providing the ability to track solution branches through turning points while maintaining robustness and accuracy. The following subsections detail these approaches and outline the specific variants implemented in the present framework.

3.4.4.1 Tangent Predictor

Among the available options, the tangent predictor is one of the most common approaches for robust continuation. It determines the prediction direction by computing the tangent to the solution branch in the combined $(\tilde{\boldsymbol{u}}, \omega)$ space.

Differentiating Equation 3.2.16 gives

$$\mathbf{0} = d\boldsymbol{G} = \nabla_{\tilde{u}}\boldsymbol{G}\,\tau_{\tilde{\boldsymbol{u}}} + \nabla_{\omega}\boldsymbol{G}\,\tau_{\omega}, \tag{3.4.3}$$

where tangent vector is denoted by $\boldsymbol{\tau} = \begin{bmatrix} \tau_{\tilde{\boldsymbol{u}}} \\ \tau_{\omega} \end{bmatrix}$. For the initialisation of the tangent direction, it is assumed that the starting point is far from a bifurcation. As a result, a unit direction is imposed on the frequency component, i.e. $\tau_{\omega,0} = 1$ for forward continuation and $\tau_{\omega,0} = -1$ for backward continuation. Under this assumption, the tangent vector is computed by solving:

$$\nabla_{\tilde{u}}\boldsymbol{G}\,\boldsymbol{\tau}_{\tilde{u},0} = -\tau_{\omega,0}\nabla_{\omega}\boldsymbol{G}. \tag{3.4.4}$$

In subsequent steps, the orientation of the solution branch is preserved by enforcing the condition $\boldsymbol{\tau}_{i-1}^T\boldsymbol{\tau}_i = 1$, which leads to the bordered system formulation which can be solve like seen in Section 3.3.1.

$$\begin{bmatrix} \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G} & \nabla_{\omega}\boldsymbol{G} \\ \boldsymbol{\tau}^{T}_{\tilde{u},i-1} & \tau_{\omega,i-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_{\tilde{u},i} \\ \tau_{\omega,i} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.4.5}$$

The $\nabla_w \boldsymbol{G}$ is approximated using finite differences. The full-rank condition from Section 3.4.1 guarantees the existence of a unique straight-line tangent to the solution branch. However, the length and orientation of the vector $\boldsymbol{\tau}$ remain undetermined. A normalisation condition must be applied to ensure uniqueness. This leads to the expression for the tangent predictor:

$$(\tilde{\boldsymbol{u}}^{p}_{i+1}, \omega^{p}_{i+1}) := (\tilde{\boldsymbol{u}}_i, \omega_i) + s_i \frac{(\boldsymbol{\tau}_{\tilde{u}}, \tau_w)}{\|\tau\|}. \tag{3.4.6}$$

Thanks to the tangent formulation, the continuation can progress across the entire solution space at each step, rather than being restricted to variations in ω as in natural parameter continuation. A detailed comparison between these continuation strategies is provided by Pan et al. [44]. The tangent predictor is illustrated in Figure 3.4.2.

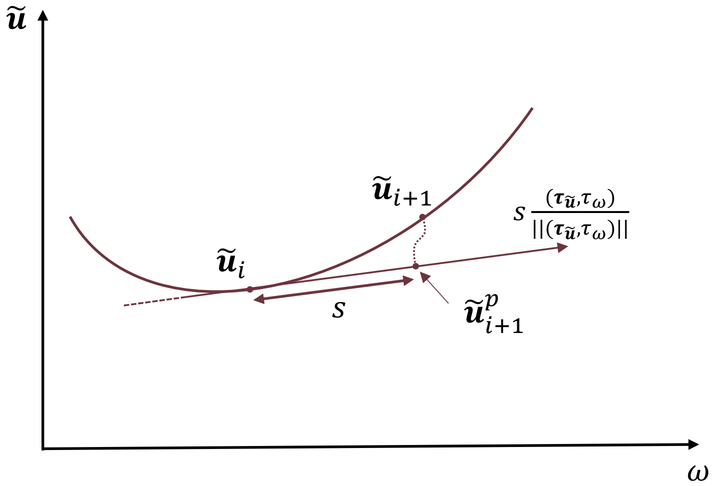


Figure 3.4.2: Prediction along the tangent. The dotted line corresponds to a hypothetical correction path, as if no correction were chosen.

3.4.4.2 Secant Predictor

While the tangent predictor exploits the local derivative of the solution branch, an alternative approach consists of using interpolation from previously computed points. This leads to the family of secant predictors, which approximate the branch by fitting a polynomial in $(\tilde{\boldsymbol{u}}, \omega)$ through a set of known solutions [79]. A polynomial of degree p in ω, interpolating $p + 1$ known points, can be constructed to serve as a predictor. When $p = 0$, the extrapolation is trivial and simply reproduces the last computed point, i.e. $\tilde{\boldsymbol{u}}^{p}_{i+1} = \tilde{\boldsymbol{u}}_i$, which is not of practical interest. Therefore, the tangent predictor is employed initially until $p+1$ solution points have been computed, allowing the construction of a higher-order secant predictor.The first-order case ($p = 1$), also known as the classical secant predictor, is illustrated in Figure 3.4.3 and defined by:

$$(\tilde{\boldsymbol{u}}^{p}_{i+1}, \omega^{p}_{i+1}) := (\tilde{\boldsymbol{u}}_i, \omega_i) + s_i \frac{(\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\ \omega_i - \omega_{i-1})}{\|(\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\ \omega_i - \omega_{i-1})\|}. \tag{3.4.7}$$

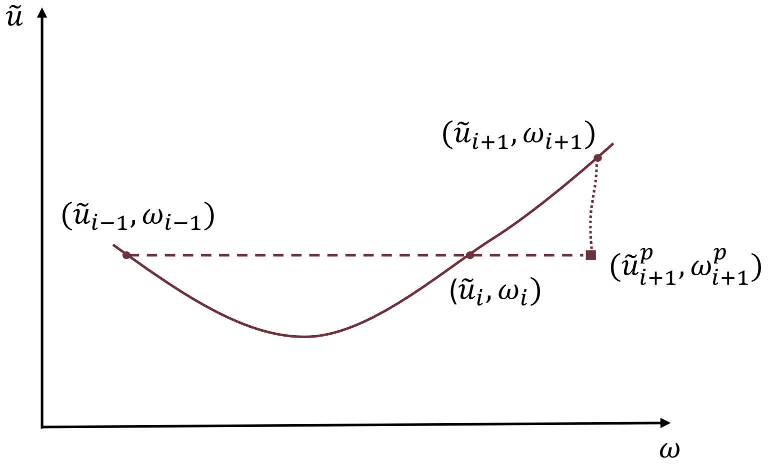


Figure 3.4.3: Secant predictor, inspired of [79], The dotted line corresponds to a hypothetical correction path.

Higher-order formulations ($p \geq 2$), some of which incorporate information from previous tangent vectors, are presented in [80]. These offer increased accuracy while remaining computationally tractable. In this thesis, the extrapolation strategy adopted throughout the continuation process corresponds to a second-order secant predictor ($p = 2$), which balances robustness and numerical efficiency. The predicted point must then be corrected to satisfy the governing equations. This is achieved by employing suitable correctors, which are introduced in the following section.

3.4.4.3 Pseudo-Arclength Method

The pseudo-arclength method was introduced by Keller [42]. The goal is to enforce orthogonality between the search space of the Newton iteration $(\tilde{u}^k, w^k)$ and the predictor vector, an additional constraint equation is introduced:

$$g(\tilde{\boldsymbol{u}}^k, \omega^k) = \boldsymbol{\tau}_{\tilde{u}}^T (\tilde{\boldsymbol{u}}^k - \tilde{\boldsymbol{u}}^p_i) + \tau_w (\omega^k - \omega^p_i) = 0, \quad g : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}. \tag{3.4.8}$$

Its gradient is given by

$$\nabla g = \begin{bmatrix} \boldsymbol{\tau}_{\tilde{u}} \\ \tau_\omega \end{bmatrix}, \tag{3.4.9}$$

by applying the Newton method for g and $\boldsymbol{G}$, the following system is obtained:

$$\begin{bmatrix} \nabla_{\tilde{\boldsymbol{u}}} \boldsymbol{G}(\tilde{\boldsymbol{u}}^k, \omega^k) & \nabla_\omega \boldsymbol{G}(\tilde{\boldsymbol{u}}^k, \omega^k) \\ \nabla_{\tilde{\boldsymbol{u}}} g^T & \nabla_w g \end{bmatrix} \begin{bmatrix} \Delta \tilde{\boldsymbol{u}}^k \\ \Delta \omega^k \end{bmatrix} = - \begin{bmatrix} \boldsymbol{G}(\tilde{\boldsymbol{u}}^k, \omega^k) \\ g(\tilde{\boldsymbol{u}}^k, \omega^k) \end{bmatrix}. \tag{3.4.10}$$

This method is illustrated in Figure 3.4.4 and is also summarised in Algorithm 2, which reflects its implementation within the Quanscient framework.

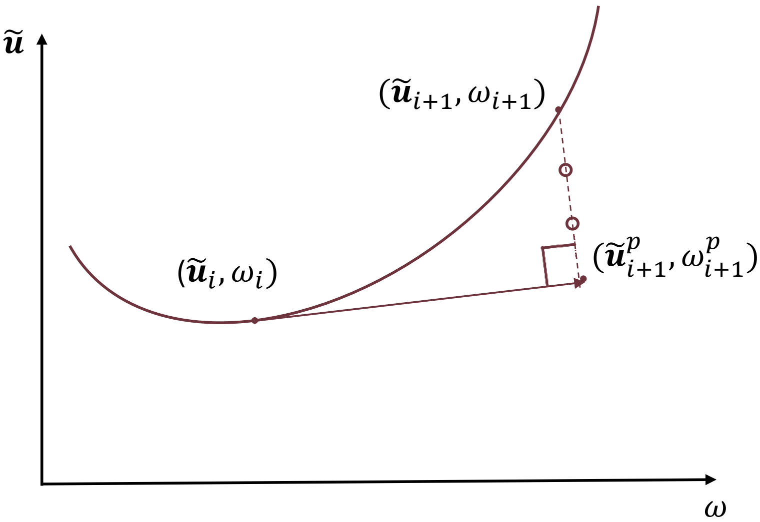


Figure 3.4.4: Illustration of the pseudo-arclength continuation method. The solution branch is shown as a solid line (-), with the predictor step indicated by an arrow (⟶). The corrector iterates $\tilde{\boldsymbol{u}}_i^k$ appear as dotted circles (∘∘), while the constraint of the corrector is represented by a dashed line (–.).

Algorithm 2 Pseudo-arclength method – Newton corrector with bordering

```
Input: $\tilde{\boldsymbol{u}}^p, \omega^p, \boldsymbol{\tau}_{\tilde{\boldsymbol{u}}}, \tau_\omega$
$\tilde{\boldsymbol{u}}^k \leftarrow \tilde{\boldsymbol{u}}^p, \quad \omega^k \leftarrow \omega^p$
iter $\leftarrow 0$
while iter $< \max_{\text{iter}}$ do
    $\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k, \boldsymbol{b}^k \leftarrow$ GenerateJacAndRHS($\tilde{\boldsymbol{u}}^k, \omega^k$)
    $\boldsymbol{G}^k \leftarrow \boldsymbol{b}^k - \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k\tilde{\boldsymbol{u}}^k$
    if $\|\boldsymbol{G}^k\| <$ tol then
        break
    end if
    $\nabla_\omega\boldsymbol{G}^k \leftarrow$ FiniteDifference($\boldsymbol{G}, \tilde{\boldsymbol{u}}^k, \omega^k, h = 0.005$)
    $g^k \leftarrow \boldsymbol{\tau}_{\tilde{\boldsymbol{u}}}^T(\tilde{\boldsymbol{u}}^k - \tilde{\boldsymbol{u}}^p) + \tau_\omega(\omega^k - \omega^p)$
    $\nabla_{\tilde{\boldsymbol{u}}}g \leftarrow \boldsymbol{\tau}_{\tilde{\boldsymbol{u}}}, \quad \nabla_\omega g \leftarrow \tau_\omega$
    $(\Delta\tilde{\boldsymbol{u}}^k, \Delta\omega^k) \leftarrow$ BorderingAlgorithm($\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}^k, \nabla_\omega\boldsymbol{G}^k, \nabla_{\tilde{\boldsymbol{u}}}g, \nabla_\omega g, \boldsymbol{G}^k, g^k$)
    $\tilde{\boldsymbol{u}}^{k+1} \leftarrow \tilde{\boldsymbol{u}}^k + \Delta\tilde{\boldsymbol{u}}^k, \quad \omega^{k+1} \leftarrow \omega^k + \Delta\omega^k$
    $\tilde{\boldsymbol{u}}^k \leftarrow \tilde{\boldsymbol{u}}^{k+1}, \quad \omega^k \leftarrow \omega^{k+1}$
    SetData($\tilde{\boldsymbol{u}}^k, \omega^k$)                                  ▷ Update fields
    iter $\leftarrow$ iter + 1
end while
Return: $\tilde{\boldsymbol{u}}^k, \omega^k$, iter
```

3.4.4.4 Arclength Method

An alternative to the pseudo-arclength formulation is the arclength method [43], which replaces the orthogonality constraint with a spherical constraint on the step size in the augmented space $(\tilde{\boldsymbol{u}}, \omega)$. This constraint is expressed as

$$g(\tilde{\boldsymbol{u}}^k, \omega^k) = \|\tilde{\boldsymbol{u}}^k - \tilde{\boldsymbol{u}}_i\|^2 + (\omega^k - \omega_i)^2 - s^2. \tag{3.4.11}$$

The solution procedure remains identical to that of the pseudo-arclength method (Equation 3.4.10), with only g and ∇g modified accordingly. This constraint is illustrated in Figure 3.4.5.

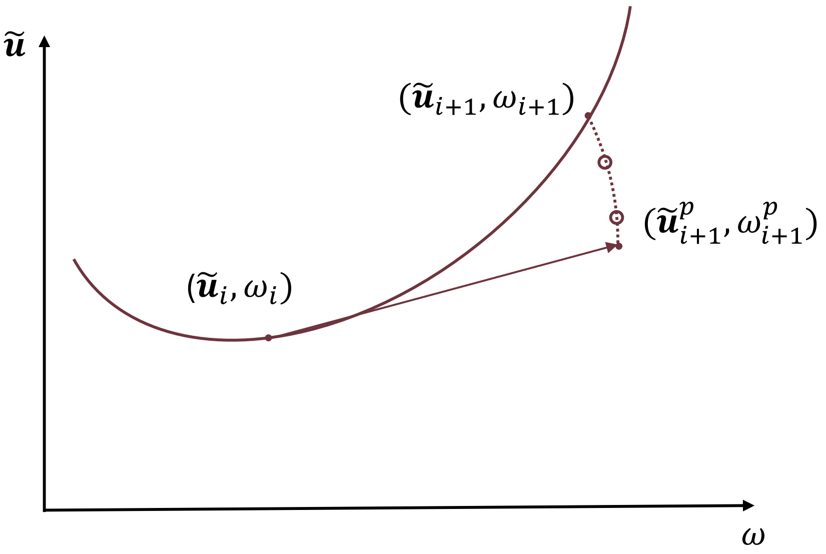
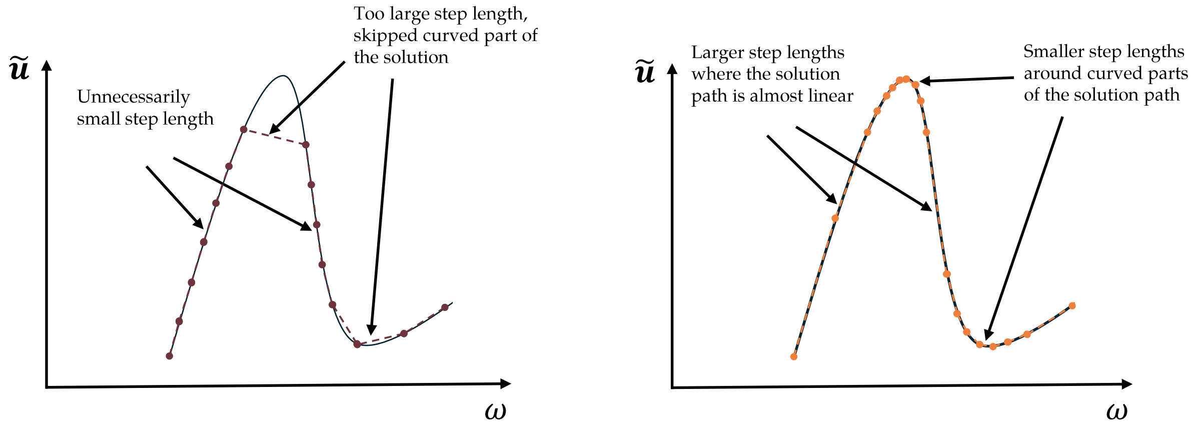


Figure 3.4.5: Illustration of the arclength continuation method. The solution branch is shown as a solid line (—), while the direction of the predictor step is indicated by an arrow (→). The corrector iterates $\tilde{\boldsymbol{u}}_i^k$ are represented by dotted circles (∞), and the constraint imposed by the corrector is depicted by a dashed line (– –).

With the direction of progression established by the chosen predictor and corrector, the final element of the continuation procedure is the control of the step length s, detailed in the following section.

3.4.5 Step Size Control

Adaptive step-size control is a key component of continuation methods, as it ensures both numerical robustness and computational efficiency. Using a constant step size may lead to failure in tracking turning points or bifurcations in regions of high curvature, while resulting in unnecessary conservatism in nearly linear segments [79]. To overcome these limitations, a variety of adaptive strategies have been proposed [81–83], all relying on a systematic adjustment of the step length based on information collected during the continuation process. This phenomenon can be observed in Figure 3.4.6, where the effectiveness of adaptive step-size control is illustrated.



(**a**) Constant step size. (**b**) Adaptive step size.

Figure 3.4.6: Comparison between constant and adaptive step-size continuation. The black curve represents the full solution branch, whereas the orange and ruby curves correspond to the solution paths traced during the iterative process inspired by [84].

Two strategies are examined in this work: the iteration-based approach, presented in Section 3.4.5.1, and the angle-based refinement, described in Section 3.4.5.2.

In general, the continuation step size s is updated at each step according to a scaling factor ξ_i, while remaining bounded within prescribed limits:

$$s_{i+1} = \min\left(s_{\max},\ \max\left(s_{\min},\ \xi_i \cdot s_i\right)\right), \tag{3.4.12}$$

where $s_{\min}$ and $s_{\max}$ denote the minimum and maximum admissible step sizes. If the updated value $\xi_i \cdot s_i$ falls below $s_{\min}$, the continuation process is considered to have failed, as the solution path becomes too difficult to follow. Conversely, if the adaptation suggests a step size larger than $s_{\max}$, the value is simply capped at this upper limit, preventing uncontrolled growth until a reduction is required. In this way, the step size evolves dynamically while remaining within acceptable bounds.

3.4.5.1 Iteration-Based Strategy

The first strategy implemented in this work is a basic iteration-based scheme, in which the scaling factor ξ_i is selected according to the convergence behaviour of the Newton corrector:

$$\xi_i \begin{cases} \xi_{up} > 1 & \text{if } N_i < N_{\max} \quad \text{(convergence achieved)}, \\ \xi_{down} < 1 & \text{if } N_i = N_{\max} \quad \text{(convergence failed)}. \end{cases}$$

In this framework, the step size is increased when the solver converges before reaching the maximum number of iterations $N_{\max}$, and decreased when convergence fails. This simple feedback mechanism provides a coarse but effective form of control.

3.4.5.2 Angle-Based Strategy

The second, more refined approach relies on an angle-based continuation criterion, as discussed by Fayezioghani et al. [84], and is favoured for its clear geometric interpretation and low computational overhead. The step size is adjusted according to the angular deviation between the predicted and corrected solution directions. This principle is closely related to the tangent-based continuation strategies commonly employed in nonlinear dynamic analyses, where the local tangent to the solution branch is already used to guide the predictor step, as illustrated in the aeroengine continuation framework of Salles et al. [85]. In this context, the angular measure provides a natural indicator of local curvature and convergence behaviour without requiring additional system evaluations.

This deviation is quantified using the angle γ_i between the connection vector ($\boldsymbol{c}_i = \tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1}, \omega_i - \omega_{i-1}$) and the tangent vector $\boldsymbol{\tau}_i$) at the current solution point. This concept is illustrated in Figure 3.4.7.

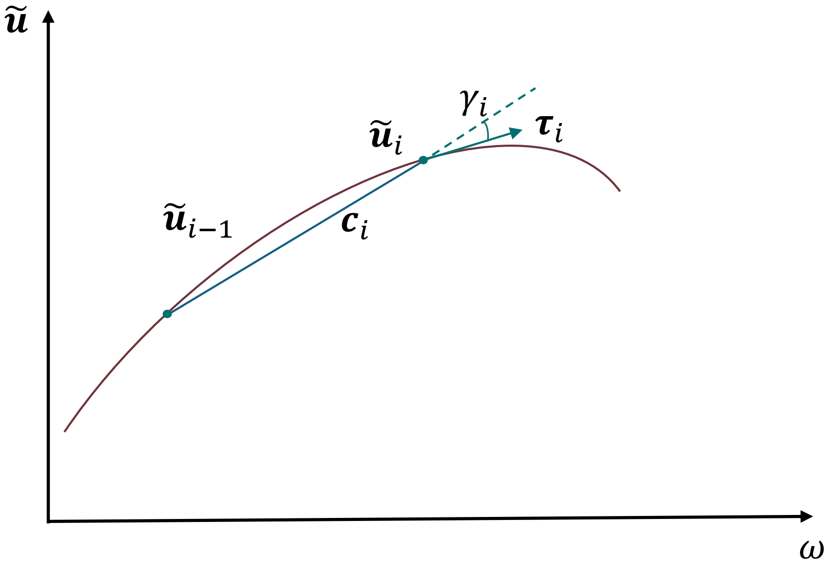


Figure 3.4.7: Adaptive step size strategy based on the angle γ_i between the connection vector $\boldsymbol{c}_i$ and the tangent vector $\boldsymbol{\tau}_i$.

The cosine of the angle γ_i is computed as:

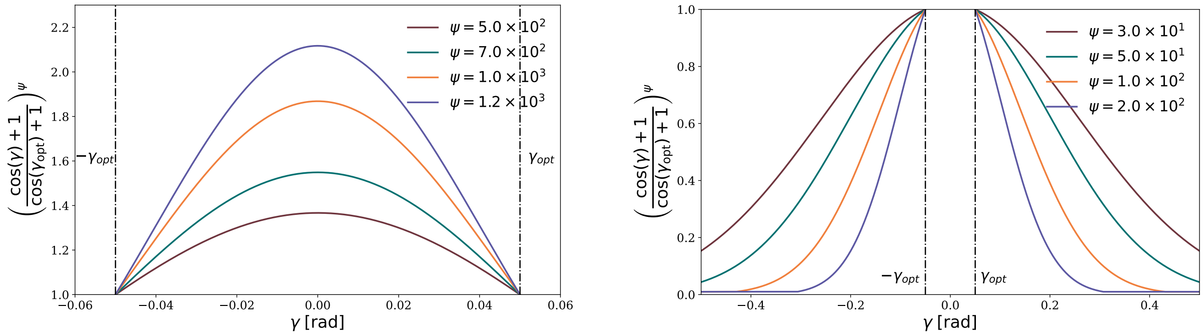
$$\cos\gamma_i = \frac{\boldsymbol{c}_i \cdot \boldsymbol{\tau}_i}{\|\boldsymbol{c}_i\|\,\|\boldsymbol{\tau}_i\|}. \tag{3.4.13}$$

Based on this measure, the scaling factor $\xi_i(\gamma_i)$ is defined as:

$$\xi_i(\gamma_i) = \left(\frac{\cos\gamma_i + 1}{\cos\gamma_{\text{opt}} + 1}\right)^{\psi}, \tag{3.4.14}$$

where γ_{opt} is a prescribed optimal angle and ψ determines the sensitivity of the adjustment. This factor directly modifies the update of s_{i+1}, making the continuation step adaptive. In particular, the step size decreases when $|\gamma_i| > |\gamma_{\text{opt}}|$ and increases when $|\gamma_i| < |\gamma_{\text{opt}}|$.

For small values of γ_{opt}, the variation of ξ_i between reduction and amplification can become excessively large. To overcome this limitation, two distinct exponents are introduced: ψ_{in}, applied when $|\gamma_i| < |\gamma_{\text{opt}}|$, and ψ_{out}, applied when $|\gamma_i| \geq |\gamma_{\text{opt}}|$. The influence of these parameters is illustrated in Figure 3.4.8. This distinction enhances the flexibility of the control strategy and prevents extreme variations in ξ_i. In addition, a lower admissible bound is imposed on ξ_i to avoid unrealistically small step sizes. When Newton's method fails to converge, the step size is directly reduced by a prescribed factor ξ_{down}. Altogether, this procedure yields a robust and efficient continuation strategy, as summarised in Algorithm 3.



(**a**) Effect of γ_{in} on ξ_i.

(**b**) Effect of γ_{out} on ξ_i.

Figure 3.4.8: Illustration of the adaptive step-size control for different values of ψ_{in} and ψ_{out}, with $\gamma_{\text{opt}} = 0.05$ rad.

Algorithm 3 Adaptive Time-Step Angle-Based Strategy.

```
Input: $\psi_{\text{in}}, \psi_{\text{out}}, \xi_{\text{min}}, \xi_{\text{down}}, \gamma_{\text{opt}}$, iter_newton, MAX_ITER
Input: Previous solution: $\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_{i-1}, \omega_i, \omega_{i-1}, \boldsymbol{\tau}_i, s_i$
Input: iter_newton
if iter_newton == MAX_ITER then
    $s_{i+1} \leftarrow \xi_{\text{down}}\, s_i$
else
    $\boldsymbol{c}_i \leftarrow \left(\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\ \omega_i - \omega_{i-1}\right)$
    $\gamma_i \leftarrow \arccos\left(\frac{\boldsymbol{c}_i \cdot \boldsymbol{\tau}_i}{\|\boldsymbol{c}_i\|\,\|\boldsymbol{\tau}_i\|}\right)$
    if $|\gamma_i| < |\gamma_{\text{opt}}|$ then
        $\psi \leftarrow \psi_{\text{in}}$
    else
        $\psi \leftarrow \psi_{\text{out}}$
    end if
    $\xi_{\text{raw}} \leftarrow \left(\frac{\cos(\gamma_i)+1}{\cos(\gamma_{\text{opt}})+1}\right)^{\psi}$
    $\xi_i \leftarrow \max\left(\xi_{\text{min}},\ \xi_{\text{raw}}\right)$
    $s_{i+1} \leftarrow \xi_i\, s_i$
end if
```

3.4.6 Continuation Loop

Once the predictor, corrector, and step size control strategies have been defined, they are integrated into the continuation loop, which governs the overall progression along the solution branch. The algorithm operates between a starting frequency ω_s, a minimum frequency ω_{min}, and a maximum frequency ω_{max}. Algorithm 4 illustrates the main loop implemented in Python using custom classes. The logic is structured around three objects: `Predictor`, `Corrector`, and `StepSize`, each encapsulating specific behaviours related to the continuation procedure. The user defines the strategy by instantiating the appropriate versions of these classes before launching the loop. The full implementation of the code can be found at `https://github.com/VictorRenkin/continuation_sparselizard.git`.

Algorithm 4 Continuation loop.

```
Input: Mesh, elasticity problem
Input: $\omega_s, \omega_{\min}, \omega_{\max}$
Input: Predictor(length_s, tan_w, order)
Input: Corrector(MAX_ITER, TOL)
Input: StepSize($s_{\min}, s_{\max}, s_{\text{start}}$, MAX_ITER)
$\omega_i \leftarrow \omega_s$
$(\tilde{\boldsymbol{u}}_i, \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}_i, \boldsymbol{G}_i) \leftarrow$ NewtonSolve($\omega_i$)    ▷ See 1.
$(\tau_{\tilde{\mathbf{u}}}, \tau_\omega) \leftarrow$ Predictor.initialise_tan($\tilde{\boldsymbol{u}}_i, \omega_i, \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}_i, \boldsymbol{G}_i$)
PreviousPoint.add(($\tilde{\boldsymbol{u}}_i, \omega_i, \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}_i, \boldsymbol{G}_i, \tau_{\tilde{\mathbf{u}}}, \tau_\omega$))
while $\omega_i \in (\omega_{\min}, \omega_{\max})$ do
    $(\tau_{\tilde{\mathbf{u}}}, \tau_\omega) \leftarrow$ Predictor.prediction_direction(PreviousPoint)
    $s \leftarrow$ StepSize.adaptif_step(PreviousPoint, Predictor)
    if $s < s_{\min}$ then
        break
    end if
    $(\tilde{\boldsymbol{u}}^p, \omega^p) \leftarrow$ Predictor.predict(PreviousPoint, $s, \tau_{\tilde{\mathbf{u}}}, \tau_\omega$)
    SetData($\tilde{\boldsymbol{u}}^p, \omega^p$)
    if $\omega^p \notin (\omega_{\min}, \omega_{\max})$ then
        break
    end if
    $(\tilde{\boldsymbol{u}}^k, \omega^k, \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}^k, \boldsymbol{G}^k$, iter$) \leftarrow$ Corrector.correct_step(Predictor, PreviousPoint)
    if iter = Max_iter then
        SetData($\tilde{\boldsymbol{u}}_i, \omega_i$)
        if $s < s_{\min}$ then
            break
        end if
    else
        $\tilde{\boldsymbol{u}}_i \leftarrow \tilde{\boldsymbol{u}}^k$
        $\omega_i \leftarrow \omega^k$
        $\nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}_i \leftarrow \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}^k$
        $\boldsymbol{G}_i \leftarrow \boldsymbol{G}^k$
        PreviousPoint.add(($\tilde{\boldsymbol{u}}_i, \omega_i, \nabla_{\tilde{\mathbf{u}}} \boldsymbol{G}_i, \boldsymbol{G}_i, \tau_{\tilde{\mathbf{u}}}, \tau_\omega$))
    end if
end while
```

3.5 Non-linear Normal Mode

With the continuation framework now fully established, attention is turned to a specific application: the computation of NNMs for conservative systems. The methodology presented in this section follows in its entirety the approach developed by Clément Grenat [1], which serves as the foundation for the present implementation.

In this framework, both the external forcing vector $\boldsymbol{f}_{\text{ext}}$ and the damping matrix $\boldsymbol{D}$ are set to zero, leading to the autonomous system

$$\mathbf{G}(\tilde{\boldsymbol{u}}, \omega) = \mathbf{Z}(\omega)\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{f}}_{\text{nl}}(\tilde{\boldsymbol{u}}, \omega) = 0. \tag{3.5.1}$$

To obtain non-trivial periodic solutions distinct from the trivial equilibrium two additional constraints are introduced. First, a phase condition is imposed to ensure the uniqueness of the periodic orbit, as detailed in Section 3.5.1. Second, a fictive energy input is applied to relax the system and initiate the motion away from equilibrium, as described in Section 3.5.2. With these two ingredients, the continuation procedure presented in Section 3.5.3 can be extended to compute NNMs in a robust and consistent manner.

3.5.1 Phase Condition

NNMs are solutions of an autonomous nonlinear system that is invariant under time translation. If $\boldsymbol{u}(t)$ is a solution, then $\boldsymbol{u}(t+\Delta t)$ is also a solution for any shift Δt. In the time domain, this property means that periodic solutions are not unique, as every point on the periodic orbit represents the same motion. This behaviour is shown in Figure 3.5.1, where the full closed trajectory corresponds to one periodic solution. To select a single representative of this orbit, a phase condition must be introduced.

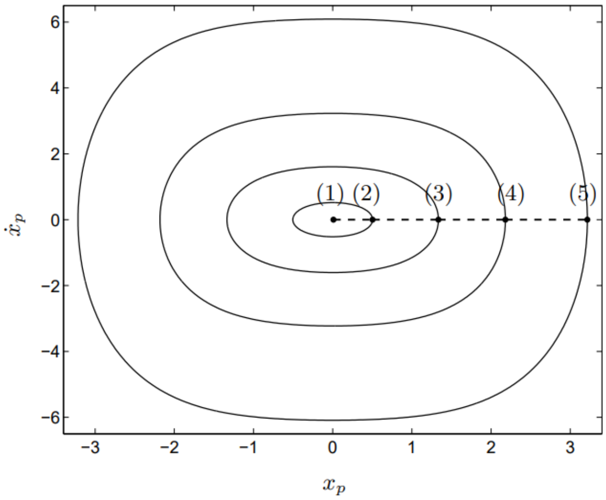


Figure 3.5.1: Family of periodic motions in the phase space. Dashed lines (—) denote the path-following curve, and solid lines (—) represent the periodic solutions obtained in the continuation of periodic motions for a Duffing oscillator [29].

Within the harmonic balance framework, the same temporal invariance manifests differently, as the time dependence is embedded in the Fourier representation [1]. The transformation associated with a phase shift Δt is expressed as

$$\begin{aligned} \boldsymbol{T}_{\tilde{\boldsymbol{u}}} &: \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, \\ &(\tilde{\boldsymbol{u}}, \Delta t) \mapsto \left(\mathrm{Rot}(\Delta t) \otimes \boldsymbol{I}_n\right) \tilde{\boldsymbol{u}}, \end{aligned} \tag{3.5.2}$$

where the rotation operator is defined as

$$\mathrm{Rot}(\Delta t) = \mathrm{diag}\left(1, \mathrm{Rot}_1(\Delta t), \ldots, \mathrm{Rot}_{n_H}(\Delta t)\right), \tag{3.5.3}$$

$$\mathrm{Rot}_j(\Delta t) = \begin{bmatrix} \cos(j\omega\Delta t) & -\sin(j\omega\Delta t) \\ \sin(j\omega\Delta t) & \cos(j\omega\Delta t) \end{bmatrix}. \tag{3.5.4}$$

The residual defined in Equation 3.5.1 is invariant under the transformation $\boldsymbol{T}_{\tilde{\boldsymbol{u}}}$. As a consequence, the Jacobian $\nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G}$ becomes singular, reflecting the fact that the phase of the periodic orbit is undetermined. To eliminate this degeneracy and recover a unique periodic solution, a phase condition $p(\tilde{\boldsymbol{u}})$ must be added to the system. Two alternatives are considered in this work: the simple phase condition and the robust phase condition[4].

3.5.1.1 Simple Phase Condition

A straightforward option consists in imposing that one selected Fourier coefficient vanishes:

$$p(\tilde{\boldsymbol{u}}) = \tilde{u}_m = 0, \tag{3.5.5}$$

where m denotes the index of the chosen harmonic component. Although easy to implement, this constraint does not minimise the change in the solution between successive continuation steps. As a result, the selected coefficient may drift along the solution branch, leading to phase inconsistencies that complicate the interpretation of the results and may degrade numerical stability, particularly near modal interactions [1].

3.5.1.2 Robust Phase Condition

To overcome these limitations, a more reliable alternative was introduced by Seydel [79]. In the time domain, the phase of the new solution is constrained relative to the previous one by

$$p(\dot{\boldsymbol{u}}, \boldsymbol{u}) = \int_0^T \dot{\boldsymbol{u}}_{i-1}^T \, \dot{\boldsymbol{u}} \, \mathrm{d}t, \tag{3.5.6}$$

which enforces orthogonality between consecutive velocity fields. Transposed into the frequency domain, this becomes

$$p(\tilde{\boldsymbol{u}}) = \tilde{\boldsymbol{u}}_{i-1}^T \, (\tilde{\nabla} \otimes \mathbb{I}_n) \, \tilde{\boldsymbol{u}}_i, \tag{3.5.7}$$

ensuring minimal phase deviation between continuation steps. In addition, the gradient $\nabla_{\tilde{\boldsymbol{u}}} p$ coincides with the singular eigenvector $\tilde{\boldsymbol{\varphi}}_1$ associated with temporal invariance in the frequency domain. Thus, the condition effectively removes the component of $\tilde{\boldsymbol{u}}$ along $\tilde{\boldsymbol{\varphi}}_1$, restoring uniqueness of the periodic solution for any values of $\tilde{\boldsymbol{u}}$ and ω [1].

3.5.2 Equation of Motion with Fictive Energy

While the phase condition ensures the uniqueness of the solution, it can still be challenging to obtain convergence in some computations. To improve robustness, Muñoz-Almaraz [86,

---

[4] The names "simple" and "robust" phase conditions are used for clarity, as no universally accepted terminology exists in the literature.

87] proposed an augmented formulation in which a non-conservative fictive energy term e_{fic}, depending on a relaxation parameter μ, is introduced. This term is added to the equations of motion to facilitate the computation of valid solutions:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \mu \boldsymbol{e}_{\text{fic}}(\dot{\boldsymbol{u}}) + \boldsymbol{K}\boldsymbol{u} + \boldsymbol{f}_{\text{nl}}(\boldsymbol{u}) = \mathbf{0}\,. \tag{3.5.8}$$

When $\mu \neq 0$, the system becomes non-conservative. For a solution to be periodic, the energy input and output must balance . The relaxation parameter μ must therefore converge to zero at a periodic solution [86]. In this case, the conservative nature of Equation 3.5.1 remains unchanged, but the system becomes well-posed. After applying the HBM, the resulting residual is:

$$\mathbf{G}(\tilde{\boldsymbol{u}}, \omega, \mu) = \mathbf{Z}(\omega)\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{f}}_{\text{nl}}(\tilde{\boldsymbol{u}}, \omega) + \mu \tilde{\boldsymbol{e}}_{\text{fic}}(\tilde{\boldsymbol{u}}) = \mathbf{0}\,. \tag{3.5.9}$$

The fictive energy $\tilde{\boldsymbol{e}}_{\text{fic}}$ must be equal to the eigenvector $\tilde{\varphi}_1$ to preserve the eigenvalues and eigenvectors of the system [1][5].

$$\tilde{\boldsymbol{e}}_{\text{fic}} = (\tilde{\nabla} \otimes \mathbb{I}_n)\,\tilde{\boldsymbol{u}}\,. \tag{3.5.10}$$

3.5.3 Continuation Scheme

Now that an equation ensuring the uniqueness of the solution has been established and the relaxation parameter defined, the continuation strategy presented in Section 3.4 can be extended to the case of NNMs. The overall approach remains based on the same principles as for the NLFR formulation, but is adapted to include the additional parameter and the extra equation introduced by the fictive energy term. Different predictor and corrector strategies are employed within this adapted framework.

3.5.3.1 Initialisation

The continuation process begins by selecting a specific linear mode of interest. The associated eigenvector, obtained from linear modal analysis, is evaluated at its resonance frequency. This eigenvector is then used to define the initial condition for the continuation, which is projected onto either the cosine or sine component of the first harmonic. Throughout this thesis, the cosine component is systematically selected. In order to prevent the continuation solver from converging towards the trivial solution, an amplitude condition is imposed for the first point. This condition enforces a prescribed continuation step size and ensures that the solution does not collapse towards zero. It is expressed in terms of the Euclidean norm of the harmonic displacement vector as

$$\|\tilde{\boldsymbol{u}}^k\| - \|\tilde{\boldsymbol{u}}_i\| + s > 0. \tag{3.5.11}$$

---

[5] If the objective is only continuation, the exact form of the fictive energy does not matter, as long as it introduces a non-conservative effect. However, during bifurcation analysis, the Jacobian of the augmented system must match $\tilde{\varphi}_1$ [1].

3.5.3.2 Predictor

The predictor step relies on the computation of a tangent direction that accounts for all active continuation parameters, including the frequency ω and any additional parameters μ. This approach is referred to as the generalised tangent predictor and is distinct from the simpler form introduced in Section 3.4.4.1, which considers only ω. The tangent vector is computed by solving the following linear system:

$$\begin{bmatrix} \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G} & \nabla_{\omega}\boldsymbol{G} & \nabla_{\mu}\boldsymbol{G} \\ \nabla_{\tilde{\boldsymbol{u}}}p & 0 & 0 \\ \boldsymbol{\tau}_{\tilde{\boldsymbol{u}},i-1} & \tau_{\omega,i-1} & \tau_{\mu,i-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_{\tilde{\boldsymbol{u}},i} \\ \tau_{\omega,i} \\ \tau_{\mu,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 0 \\ 1 \end{bmatrix} . \tag{3.5.12}$$

Based on this direction, the predictor is constructed using a step of size s as:

$$(\tilde{\boldsymbol{u}}^p_{i+1}, \omega^p_{i+1}, \mu^p_{i+1}) := (\tilde{\boldsymbol{u}}_i, \omega_i, \mu_i) + s\frac{(\tau_{\tilde{\boldsymbol{u}},i}, \tau_{\omega,i}, \tau_{\mu,i})}{\|\boldsymbol{\tau}_i\|} . \tag{3.5.13}$$

Alternatively, a secant predictor may be used. This method does not require explicit computation of a tangent and is defined as:

$$(\tilde{\boldsymbol{u}}^p_{i+1}, \omega^p_{i+1}, \mu^p_{i+1}) = (\tilde{\boldsymbol{u}}_i, \omega_i, \mu_i) + s_i\frac{(\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\ \omega_i - \omega_{i-1},\ \mu_i - \mu_{i-1})}{\left\|(\tilde{\boldsymbol{u}}_i - \tilde{\boldsymbol{u}}_{i-1},\ \omega_i - \omega_{i-1},\ \mu_i - \mu_{i-1})\right\|} . \tag{3.5.14}$$

3.5.3.3 Corrector

The correction step refines the predicted solution by solving an augmented system that incorporates the residual $\boldsymbol{G}$, the phase condition p, and a continuation constraint g. In the pseudo-arclength formulation, this constraint enforces orthogonality with the previous tangent:

$$g(\tilde{\boldsymbol{u}}^k, \omega^k, \mu^k) = \tau^T_{\tilde{\boldsymbol{u}},i}(\tilde{\boldsymbol{u}}^k - \tilde{\boldsymbol{u}}^p_i) + \tau_{\omega,i}(\omega^k - \omega^p_i) + \tau_{\mu,i}(\mu^k - \mu^p_i) = 0 . \tag{3.5.15}$$

Alternatively, a spherical constraint may be adopted:

$$g(\tilde{\boldsymbol{u}}^k, \omega^k, \mu^k) = \|\tilde{\boldsymbol{u}}^k - \tilde{\boldsymbol{u}}_i\|^2 + (\omega^k - \omega_i)^2 + (\mu^k - \mu_i)^2 - s^2 . \tag{3.5.16}$$

Both variants result in the same Newton-type linear system:

$$\begin{bmatrix} \nabla_{\tilde{\boldsymbol{u}}}\boldsymbol{G} & \nabla_{\omega}\boldsymbol{G} & \nabla_{\mu}\boldsymbol{G} \\ \nabla_{\tilde{\boldsymbol{u}}}p & 0 & 0 \\ \nabla_{\tilde{\boldsymbol{u}}}g & \nabla_{\omega}g & \nabla_{\mu}g \end{bmatrix} \begin{bmatrix} \Delta\tilde{\boldsymbol{u}} \\ \Delta\omega \\ \Delta\mu \end{bmatrix} = - \begin{bmatrix} \boldsymbol{G} \\ p \\ g \end{bmatrix} . \tag{3.5.17}$$

With the continuation strategies for both the NLFRs and NNMs formulations now fully established, the next step is to examine the software environment used to compute the systems under study, including cases with a large number of degrees of freedom.

3.6 QUANSCIENT ALLSOLVE Software

QUANSCIENT ALLSOLVE is a cloud-native multiphysics simulation platform designed to provide engineers and researchers with fast, scalable, and flexible simulation capabilities directly through a web browser [88] with serveur work with the Amazon Web Services (AWS). Established in 2021 and headquartered in Tampere, Finland, Quanscient has developed this platform to address the limitations of traditional simulation tools by leveraging advanced cloud computing and quantum algorithms.

- **Cloud Scaling:** The platform offers virtually unlimited computational power and RAM on any device, significantly reducing simulation runtimes from days to mere hours or minutes. This scalability supports unlimited parallel simulations, enabling extensive parametric sweeps and optimisation tasks [88].
- **Python Scripting Interface:** The platform provides automation and flexibility for setting up and running simulations programmatically [88], which will be employed throughout this thesis.
- **Numerical Methods:** The simulation core relies on the FEM, combined with a DDM to enable parallel computation in a cloud environment [88].
- **Quantum Algorithms:** A quantum-native approach is being developed using the Quantum Lattice Boltzmann Method for fluid dynamics. A one-dimensional test case has already been successfully run on real quantum hardware, with further developments planned in higher dimensions [88].

This overview highlights the main features of QUANSCIENT ALLSOLVE. The analysis is then divided into two parts. First, the underlying source code is examined in Section 3.6.1, with particular attention to the external packages on which the platform relies. Second, the practical usage of the software is illustrated in Section 3.6.2, where the steps required to set up and solve a problem are presented, from the solver initialisation to the generation of the system matrices.

3.6.1 Source Code

The core of QUANSCIENT ALLSOLVE is SPARSELIZARD [89], developed by Alexandre Halbach, co-founder of Quanscient. SPARSELIZARD is an open-source C++ FEM library relies on several external packages, as listed in Table 3.6.1.

External Library	Description
OPENBLAS [90]	Provides optimised and multithreaded operations for dense matrices and vectors.
PETSC [57]	A parallel iterative solver for large sparse systems.
MUMPS [55]	A parallel direct solver for large sparse systems.
SLEPC [91]	Used in conjunction with PETSc to solve large-scale sparse eigenvalue problems.
GMSH [92]	Generate meshes used as input for the FEM simulations.

Table 3.6.1: External libraries used by SPARSELIZARD.

3.6.2 Usage

This section provides a concise overview of how Quanscient Allsolve is employed in practice to set up and solve a physical problem.

3.6.2.1 Initialisation

The first step is to select the type of solver to be used, either a direct solver or an iterative solver. In this thesis, the direct solver was consistently employed, relying on NUMPs. The solver precision must also be specified, choosing between double precision and quadruple precision.

Subsequently, the number of nodes must be defined, which determines the partitioning of the discretised domain. Finally, the node type is selected, specifying the memory available per node. Each core provides 16 GB of memory, with configurations available up to 64 cores.

3.6.2.2 Define the Mesh

Before solving a problem, the computational mesh must first be defined and prepared. While Quanscient Allsolve supports internal mesh generation, it is often preferable to import an external mesh, for example one created using Gmsh.

In such cases, a mesh object is first initialised, and the relevant physical regions are registered. A dedicated region is then reserved to represent the external boundary referred to as the *skin* of the domain. This skin must be selected in advance so that Quanscient can automatically detect and associate the corresponding boundary elements during mesh import. The mesh is then partitioned to support parallel computation. Finally, the mesh file is loaded and all entities including the skin are mapped accordingly, ensuring consistent boundary identification and preparation for distributed simulation.

```
mesh.mesh = qs.mesh()
mesh.mesh.setphysicalregions(*reg.get_region_data())
mesh.skin = reg.get_next_free()
mesh.mesh.selectskin(mesh.skin)
mesh.mesh.partition()
mesh.mesh.load("gmsh:simulation.msh", mesh.skin, 1, 1)
```

3.6.2.3 Define the Field

In this context, a field represents the physical unknown to be computed over the domain in this case, the displacement field. A multi-harmonic formulation is adopted, where each harmonic corresponds to a pair of Fourier coefficients. For example, the following command:

```
qs.field("h1xyz", [1, 2, 3])
```

defines a field of the form:

$$\boldsymbol{u}(t) = \tilde{\boldsymbol{u}}_0 + \tilde{\boldsymbol{u}}_1^s \sin(\omega t) + \tilde{\boldsymbol{u}}_1^c \cos(\omega t).$$

Once the field has been defined, the interpolation order of the FEM approximation is specified using the `setorder` method. For instance, setting a second-order interpolation in the volume region is done as follows:

```
fld.u.setorder(PHYSREG_VOLUME, 2)
```

This improves the solution accuracy, particularly in regions with strong gradients or geometric complexity.

Finally, Dirichlet boundary conditions are imposed on selected physical regions using the `setconstraint` method. For example:

```
fld.u.setconstraint(PHYSREG_CONSTRAINT)
```

This command enforces prescribed displacements or symmetry on the region defined by `PHYSREG_CONSTRAINT`.

3.6.2.4 Define the problem

The fundamental step is to define the problem to be solved. It must be written in weak form. In this case, the motion problem is considered, and its weak formulation is given in Equation 3.1.16.

The problem is built by assembling each term over a specific physical region and using the field previously defined. The formulation object is used to collect these terms:

```
elasticity = qs.formulation()

Mddotu = -rho * qs.dtdt(qs.dof(fld.u)) * qs.tf(fld.u)
elasticity += qs.integral(PHYSREG_VOLUME, Mddotu)

Cdotu = -alpha * rho * qs.dt(qs.dof(fld.u)) * qs.tf(fld.u)
elasticity += qs.integral(PHYSREG_VOLUME, Cdotu)

Ku=qs.predefinedelasticity(qs.dof(fld.u), qs.tf(fld.u), fld.u, par.H(), 0)
elasticity += qs.integral(PHYSREG_VOLUME, Ku)

Force_apply = qs.array1x3(0,0,-200) * qs.tf(fld.u.harmonic(3))
elasticity += qs.integral(PHYSREG_LOAD_POINT, Force_apply)
```

3.6.2.5 Generating the Solver Quantities

After defining the problem, the fundamental frequency must be set:

```
qs.setfundamentalfrequency(freq)
```

It is also possible to manually assign a solution estimate to the field, for instance during continuation:

```
field_u.setdata(PHYSREG_U, u_pred)
```

Once both the frequency and the field are set, the system is assembled with:

```
elasticity.generate()
```

This generates the discretised system in the form:

$$\boldsymbol{H}\boldsymbol{x} = \boldsymbol{q},$$

where $\boldsymbol{H}$ is the assembled operator, $\boldsymbol{x}$ the vector of unknowns, and $\boldsymbol{q}$ the forcing vector.

The Jacobian matrix and the updated right-hand side can then be extracted using:

```
Jac = elasticity.A(False, True)
b = elasticity.b(False, True, True)
```

In this context, the Jacobian is denoted $\boldsymbol{J}$, and $\boldsymbol{b}$ is defined as:

$$\boldsymbol{b} = \boldsymbol{q} + (\boldsymbol{J} - \boldsymbol{H})\boldsymbol{x}.$$

The residual of the nonlinear system becomes:

$$\boldsymbol{G} = \boldsymbol{q} - \boldsymbol{H}\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{J}\boldsymbol{x}. \tag{3.6.1}$$

This residual $\boldsymbol{G}$ is used within Newton's method to iteratively correct the solution until convergence.

Given that large-scale simulations in Quanscient Allsolve can be challenging to debug, pyHarm is used as a complementary tool for rapid prototyping and straightforward troubleshooting.

3.7 pyHarm — Harmonic Balance Method Based Solver

pyHarm is a Python-based solver for nonlinear vibration analysis, built around the HBM [93]. Developed and maintained by Safran Tech, it features a modular and transparent architecture that facilitates the construction, modification, and validation of mechanical models. Designed for academic and moderately sized systems, it enables efficient model development and error identification. In the long term, the solver is expected to be extended to larger industrial applications through model-reduction strategies, in particular by integrating techniques developed within the Morphe framework, thereby bridging the gap between exploratory academic tools and scalable engineering solutions.

The following discussion first outlines the general architecture and functionality of pyHarm in Section 3.7.1, before turning to the developments carried out in this work in Section 3.7.2, which extend its scope to conservative systems and the computation of NNMs.

3.7.1 Architecture and Functionality

The core idea behind pyHarm is to treat mechanical systems as assemblies of elementary components. Each component such as a mass, spring, or nonlinear connector contributes independently to the residual vector and its Jacobian. These are assembled into a global algebraic system, formulated in the frequency domain via truncated Fourier series.

Models are defined using Python dictionaries that specify:

- the linear dynamics through mass, damping, and stiffness matrices $\boldsymbol{M}$, $\boldsymbol{D}$, and $\boldsymbol{K}$;
- the nonlinear restoring forces (e.g., cubic stiffness, hardening springs);
- the number of harmonics to be retained in the frequency-domain formulation.
- corrector, predictor, type analysis

From this input, pyHarm automatically constructs the residual vector $\boldsymbol{G}(\tilde{\boldsymbol{u}}, \omega)$ and its Jacobian. These quantities serve as the foundation for applying continuation methods in the frequency domain. The solver natively supports the computation of NLFRs curve and includes various predictor–corrector schemes, such as tangent and secant predictors combined with arclength or pseudo-arclength correctors.

3.7.2 Contributions of This Work

The objective of this work was to extend the functionality of pyHarm to enable the analysis of NNMs for conservative systems. This required the implementation of several key features:

- A dedicated `PhaseCondition` class, allowing for the enforcement of phase constraints in the absence of external forcing;
- A continuation framework tailored to the computation of backbone curves, enabling systematic tracing of NNMs across amplitude;
- The integration of multiple predictor–corrector strategies adapted to unforced conservative dynamics, including tangent and secant predictors, as well as classical and pseudo-arclength correctors.

These developments were fully integrated into the modular structure of pyHarm, preserving compatibility with existing components and workflows.

3.8 Conclusion

The methodology presented in this chapter establishes a comprehensive framework for the nonlinear vibration analysis of the considered structures. The physical modelling was formulated directly within the FEM framework, with the appropriate boundary conditions applied and the material behaviour described by a Saint Venant–Kirchhoff constitutive law under Neumann conditions, while dissipative effects were represented through Rayleigh damping. The formulation was developed in a rotating reference frame, allowing centrifugal effects to be consistently taken into account and enabling the investigation of rotation-dependent dynamic phenomena, including the construction of Campbell diagrams. The resulting nonlinear FEM model was subsequently recast within the HBM framework, providing access to steady-state periodic solutions. This led to a unified residual formulation, applicable both to forced responses, such as NLFRs, and to conservative dynamics through the computation of NNMs.

After establishing this formulation, attention can be turned to the analysis of the nonlinear system. The resolution of the nonlinear residual equation is carried out using a Newton–Raphson iterative scheme, with the resulting linear systems efficiently handled through a bordering algorithm. The evaluation of the nonlinear force vector and its Jacobian can be performed either analytically or by means of the AFT method, depending on the complexity of the nonlinear terms.

Subsequently, various predictor–corrector schemes were examined, including the tangent, secant, pseudo-arclength, and arclength methods, together with several adaptive step-size strategies. Algorithm 4 was proposed as the main continuation loop, in which all predictors, correctors, and step-size controllers were implemented as modular classes. This structure allows the straightforward inclusion of any existing or newly developed strategy that the user wishes to employ.

The methodology was then extended to the computation of NNMs conservatif. In this case, a phase condition was introduced to ensure the uniqueness of the solution, while a relaxation parameter associated with a fictive energy term was added to reduce the stiffness of the system and improve numerical convergence.

Finally, the computational tools supporting this work were presented. Quanscient Allsolve was first employed for large-scale simulations, taking advantage of its high-performance parallel capabilities. Subsequently, pyHarm was used to implement the NNMs computations and to facilitate the identification and understanding of the modelling or numerical issues encountered. This sequential use of the two tools provided both the computational power required for complex analyses and the flexibility needed for targeted developments and debugging.

4

RESULTS – PYHARM

The pyHarm software was used to test the methodology proposed by Clément Grenat [1] on simple benchmark systems. First, a single DOF oscillator is examined in Section 4.1, followed by a two DOFs oscillator in Section 4.2. These cases provide a consistent framework to validate and illustrate the effectiveness of the proposed approach.

4.1 Single Degree of Freedom System

The system analysed for the 1-glsdof case is defined by the following equation:

$$\ddot{u} + \dot{u} + u + u^2 + u^3 = F_{ext} \tag{4.1.1}$$

The backbone curve and the NLFRs are shown in Figure 4.1.1. The NNM accurately follows the response curve. A slight shift can be observed, which is attributed to the presence of damping for the analyse of NLFRs.

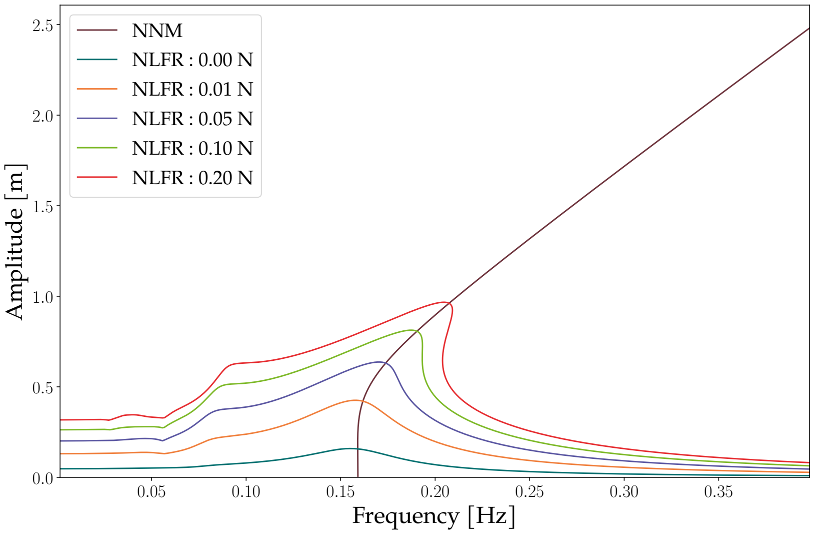


Figure 4.1.1: Comparison between the NLFRs and the NNM for the 1-DOF system exhibiting cubic and quadratic nonlinearities, as defined in Equation 4.1.1.

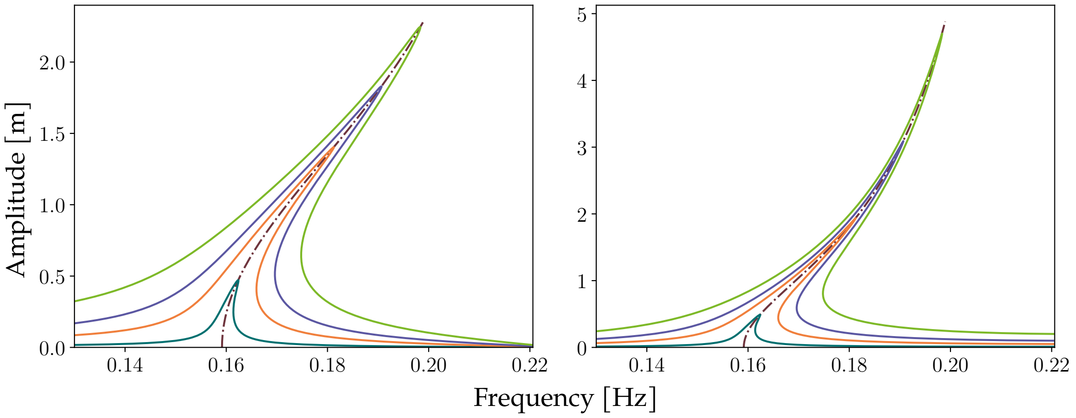
4.2 Two Degrees of Freedom System

The system under consideration is identical to the one studied in the work of Kerschen [23]. Its equations of motion are given by:

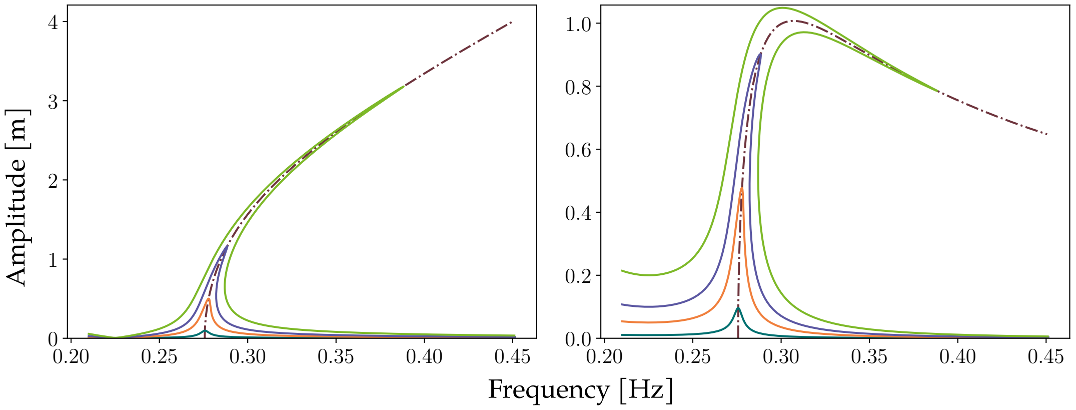
$$\ddot{u}_1 + (0.02\dot{u}_1 - 0.01\dot{u}_2) + (2u_1 - u_2) + 0.5u_1^3 = F_{\text{ext}} \tag{4.2.1}$$

$$\ddot{u}_2 + (0.02\dot{u}_2 - 0.01\dot{u}_1) + (u_2 - u_1) = 0 \tag{4.2.2}$$

Figure 4.2.1a illustrates the frequency response of the coordinate u_1, while Figure 4.2.1b shows the corresponding response for u_2. The numerical results closely match those reported by Kerschen.



(a) Near first resonance.



(b) Near Second resonance.

Figure 4.2.1: NLFRs and backbone of the system Equation 4.2.1. Left plots show u_1 and right plots u_2. Forcing amplitudes are indicated as ■ 0.01, ■ 0.05, ■ 0.1, ■ 0.2, while the backbone (NNMs) is shown in ■.

4.3 Conclusion

The computation of the NNMs was successfully achieved, confirming that the methodology proposed by Clément Grenat [1] can be effectively applied. The results emphasise the importance of a well-defined system configuration, particularly regarding the initial condition of the tangent vector τ_0, to avoid an undesired downward trend leading to the trivial zero solution. It was also observed that the relaxation parameter should initially be set to $\mu_0 = 0$ to ensure stable convergence. With this validation on a simple case, the approach can now be considered for application to systems with higher degrees of freedom, for which QUANSCIENT ALLSOLVE will be employed, as presented in the next chapter.

5

RESULTS – QUANSCIENT ALLSOLVE

The numerical experiments conducted with pyHarm in Chapter 4 provided a first validation of the methodology on small problems. Building on these results, the focus of this chapter is shifted towards large-scale computations carried out with Quanscient Allsolve, which represents the central objective of this thesis. The aim is to demonstrate the applicability, efficiency, and scalability of the proposed framework when applied to realistic FEM models of increasing complexity. This chapter first presents the hardware configuration used for the computations in Section 5.1. Two benchmark cases, namely a clamped–clamped beam and a cantilever beam, are then introduced in Section 5.2. These academic examples are used to illustrate the proposed methodology and to provide a reference for comparison. The influence of step-size control and predictor-corrector strategies is subsequently analysed in Section 5.3, with particular attention paid to their effects on robustness and convergence. The rotating frame formulation is validated in Section 5.4. Scalability aspects are then investigated in Section 5.5, with a focus on strong scaling performance for both small and large-scale test cases. Finally, the methodology is applied to an industrial case study in Section 5.6, demonstrating its relevance and potential for real-world nonlinear vibration analysis.

5.1 Hardware Overview

The numerical results presented in this work were obtained using high-performance computing resources, as the benefits of distributed-memory parallelism especially when using MPI-based solvers can only be fully realised when deploying a sufficiently large number of processors. This exceeds the capabilities of standard desktop environments and typically necessitates cloud clusters or supercomputing infrastructure. All simulations were performed on Quanscient's cloud platform, which internally relies on AWS. In particular, `r6i.large` instances were used. Each instance provides:

- 2 virtual CPU (vCPUs), corresponding to one physical core with simultaneous multithreading,
- 16 GB of RAM,
- Intel Xeon "Ice Lake" scalable processors (up to 3.5 GHz turbo),
- high-speed virtual interconnect with up to 12.5 Gbps bandwidth.

The number of `r6i.large` instances can be scaled without restriction, allowing the system to adapt dynamically to the computational load. While cloud-based solutions may not match the peak performance of on-premises supercomputers, they offer considerable flexibility and scalability, making them particularly suitable for the development and testing of distributed numerical algorithms. The effectiveness of this platform under increasing computational loads will be assessed through a dedicated scalability study presented in Section 5.5.

5.2 Testing Case

Beams are commonly used as academic benchmarks in vibration analysis because of their simple geometry and the availability of numerous analytical solutions in the literature, which makes them particularly suitable for the validation of new numerical methods [13]. In the present work, two standard configurations are considered: the clamped–clamped beam introduced in Section 5.2.1 and the cantilever beam discussed in Section 5.2.2. For each test case, the analysis is focused on the dynamics in the vicinity of the first flap-wise bending mode, which constitutes the targeted resonance of interest. All results are presented at 0 RPM. Consequently, the eigenfrequencies of the modes located close to this first flexural mode are examined in order to assess the potential occurrence of modal interactions. In addition, the backbone curve is analysed to further validate the results and to ensure the consistency of the identified nonlinear dynamics. Since the structure is externally excited for the computation of the NLFRs, the associated mode shapes are also analysed in order to identify which modes are effectively activated. The modes are denoted using the following convention: Flap-wise Bending Mode (F), Edge-wise Bending Mode (E), and Torsion Mode (T).

The discretisation employed in the present work relies on a finite element formulation based on hierarchical shape functions, as described in Section 3.1.3. The FEM used in the numerical model are Hexahedral element (HEX) and Tetrahedral element (TET), whose approximation spaces are characterised by the polynomial order P.

5.2.1 Clamped–Clamped Beam

The implementation is validated using a clamped–clamped beam, as illustrated in Figure 5.2.1, where the depicted force $\tilde{\boldsymbol{f}}_1^c$is only considered in the NLFRs section. The beam dimensions are $0.03{\times}0.03{\times}1$ m, and it is discretised using a $2{\times}2{\times}16$ mesh of HEX-P2 elements, resulting in a total of 825 nodes. The corresponding physical properties are provided in Table 5.2.1 the apply force is represented in the previous section.

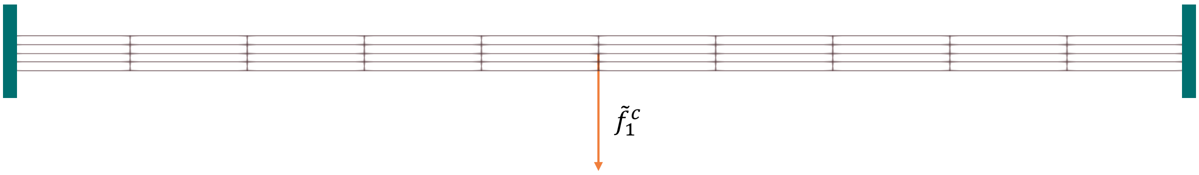


Figure 5.2.1: Geometry of the clamped–clamped beam. A nodal excitation is applied at the centre of the beam on the first cosine harmonic coefficient $\tilde{\boldsymbol{f}}_c^1$.

Parameter	**Unit**	**Value**
Density	$[\mathrm{kg\,m^{-3}}]$	$7,800$
Young's modulus	$[\mathrm{N\,m^{-2}}]$	2.1×10^{11}
Poisson's ratio	[–]	0.3
Damping matrix	[–]	$\boldsymbol{D} = 3 \times \boldsymbol{M}$

Table 5.2.1: Physical parameters for the clamped–clamped beam test case.

5.2.1.1 Mode Shapes and Eigenfrequencies

The first three modee igenfrequencies are listed in Table 5.2.2, and the corresponding mode shapes are shown in Figure 5.2.2. A 5:1 ratio exists between 1F and 3F, but the frequency separation is such that no coupling is expected in the operating range considered in this thesis.

Mode	1F	2F	3F
Frequency [Hz]	160.75	440.53	857.22
Ratio f_n/f_{1F} [-]	1.00	2.74	5.33

Table 5.2.2: Flap-wise bending eigenfrequencies of the clamped–clamped beam and their ratios relative to 1F

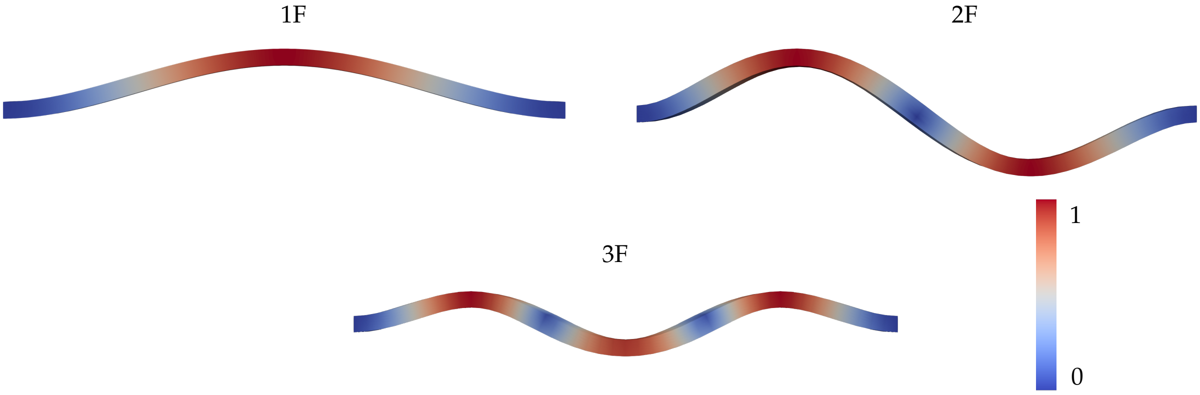


Figure 5.2.2: Mode shapes of the first three flap-wise bending of the clamped–clamped beam.

5.2.1.2 Nonlinear Frequency Response

The NLFR was computed for a cosine excitation at the first harmonic, $\tilde{\boldsymbol{f}}_1^c$, applied at the beam centre, as shown in Figure 5.2.1. This excitation point coincides with a vibration node of the 2F mode, resulting in no contribution from this mode. The response is therefore governed solely by the 1F mode, as the 3F mode is too far in frequency to be activated under the considered conditions.

The simulation parameters, including numerical settings, harmonic content, and excitation properties, are listed in Table 5.2.3. The computed NLFRs are presented in Figure 5.2.3, with an additional zoom near the bifurcation point to highlight the predictor behaviour. In this region, the choice of step-size control proves critical, and a finer adjustment could improve the convergence efficiency. A distinct hardening behaviour is observed, caused by mid-plane stretching of the beam induced by the constraints of the boundary conditions. The total computation time was 41 minutes and 7 s.

Parameter	**Value**
Newton tolerance	10^{-6}
Maximum Newton steps	10
Initial continuation step	0.5
Minimum continuation step	10^{-6}
Maximum continuation step	1
Step increase factor	1.1
Step decrease factor	0.4
Excitation amplitude	200 N
System harmonics	H_{0123}
Number of unknowns	$12,375$
Predictor type	Tangent
Corrector type	Pseudo-arclength

Table 5.2.3: Simulation parameters for the NLFR computation of the clamped–clamped beam, including numerical settings, harmonic content, and excitation properties.

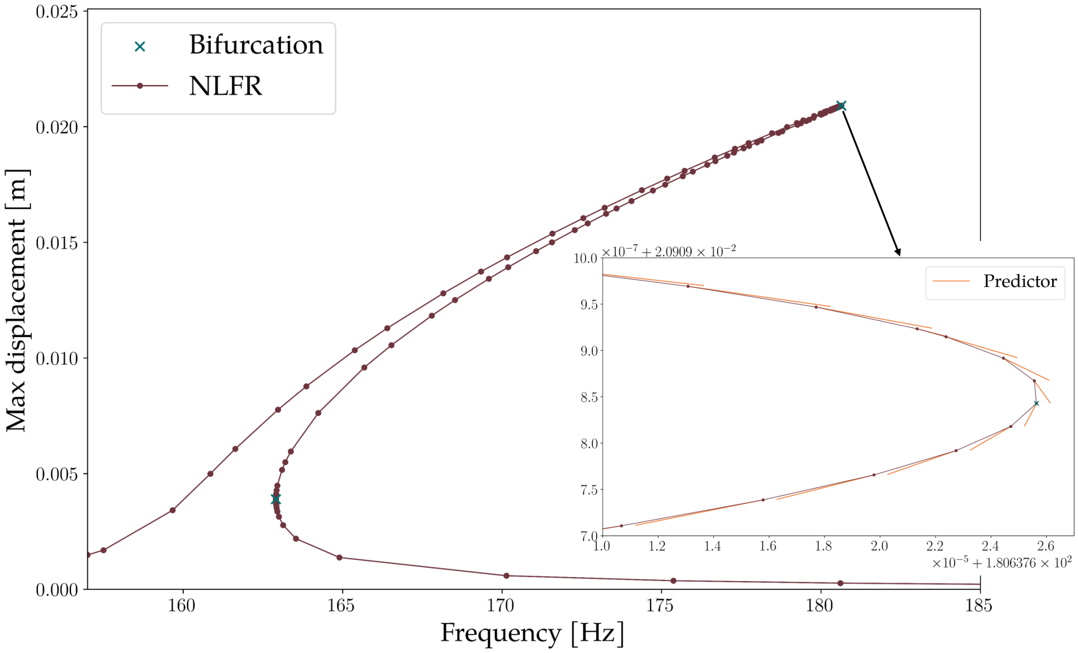


Figure 5.2.3: NLFRs of the clamped–clamped beam, with a zoom on the resonance region.

5.2.1.3 Nonlinear Frequency Response vs Backbone Curve

The first NNM of the system was computed in the conservative setting, using the numerical parameters listed in Table 5.2.4. The resulting backbone curve is compared to the peak amplitudes of the NLFR in Figure 5.2.4. Their close agreement confirms the quality of the results.

Parameter	Value
Newton tolerance	10^{-4}
Maximum Newton steps	20
Initial continuation step	5×10^{-3}
Minimum continuation step	10^{-6}
Maximum continuation step	1.1
Step increase factor	1.2
Step decrease factor	0.4
System harmonics	H_{0123}
Number of unknowns	$12,375$

Table 5.2.4: Numerical parameters used for the NNM computation of the clamped–clamped beam.

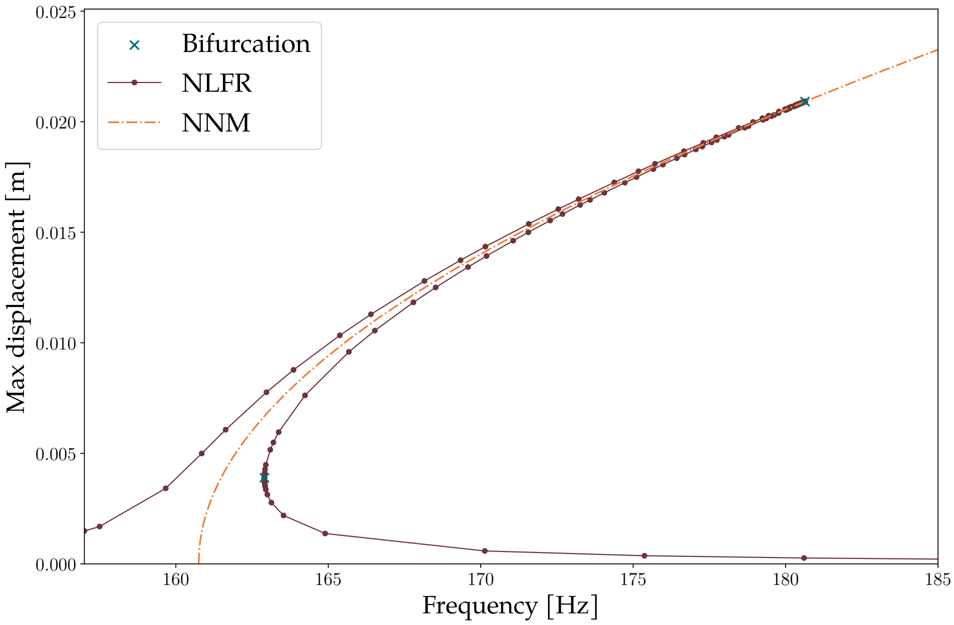


Figure 5.2.4: Comparison between the backbone curve and the NLFRs for the clamped-clamped beam.

5.2.2 Cantilever beam

A second benchmark considers a cantilever beam, which exhibits larger vibration amplitudes and a smoother variation in its frequency response compared to the clamped–clamped configuration. Accurately capturing its nonlinear behaviour requires a higher number of harmonics, making it an appropriate case to evaluate the influence of harmonic truncation on the solution accuracy. This configuration also serves as a direct comparison with the results reported by Blahoš [13], obtained using the same beam properties.

The mesh used for the simulation is shown in Figure 5.2.5 and corresponds to dimensions of $0.005 \times 0.1 \times 1$ m. A discretisation of $1 \times 8 \times 40$ HEX-P2 elements was applied, resulting in a total of $6,885$ nodes. The beam parameters are listed in Table 5.2.5, and the applied force, used for the NLFR analysis, is detailed later in this section.

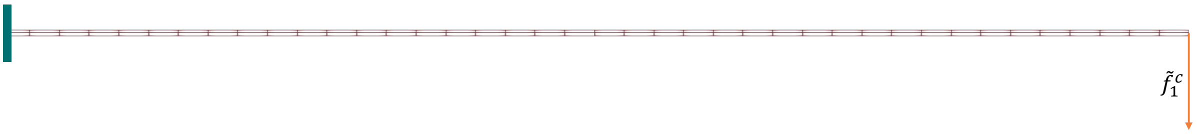


Figure 5.2.5: Diagram of the cantilever beam. A single nodal force on the first harmonic component $\tilde{\boldsymbol{f}}_1^c$ is applied at the free end of the beam, centred along its width.

Parameter	Unit	Value
Density	$[\mathrm{kg\,m^{-3}}]$	$4,400$
Young's modulus	$[\mathrm{N\,m^{-2}}]$	1.04×10^{11}
Poisson's ratio	[–]	0.3
Damping matrix	[–]	$\boldsymbol{D} = 0.2467 \times \boldsymbol{M}$

Table 5.2.5: Cantilever beam test case parameters.

5.2.2.1 Mode Shapes and Eigenfrequencies

The eigenfrequencies and the corresponding flexural mode shapes are reported in Table 5.2.6 and Figure 5.2.6. The frequency ratio of approximately 6:1 between the 2F and the 1F indicates the potential occurrence of modal coupling.

Mode	1F	2F	3F
Frequency [Hz]	3.96	24.80	69.55
Ratio f_n/f_{1F} [-]	1.00	6.27	17.56

Table 5.2.6: Eigenfrequencies of the cantilever beam and ratios relative to the first flap-wise bending mode.

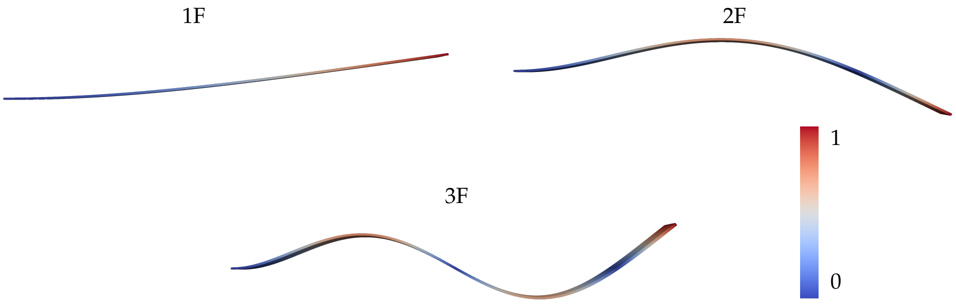


Figure 5.2.6: Bending mode shapes of the cantilever beam.

5.2.2.2 Nonlinear Frequency Response

A comparison with the results of Blahoš [13] is presented in Figure 5.2.7, using the simulation parameters reported in Table 5.2.7. A single nodal excitation force was applied to the cosine component of the first harmonic $\tilde{\boldsymbol{f}}_1^c$, positioned at the free end of the beam and centred along the cross-section width (see Figure 5.2.5). Blahoš investigated the influence of the number of harmonics retained in the analysis and the resulting modal couplings for the cantilever configuration. This aspect was not pursued in the present thesis. Nevertheless, it can be noted that the 2F mode is likely to couple with the 1F mode, underlining the importance of including at least the second harmonic for an accurate representation of this behaviour. Compared to the clamped–clamped configuration, the nonlinear hardening effect is less pronounced, as the weaker axial stretching induced by the cantilever boundary conditions reduces the geometric stiffening contribution. This attenuation of the stiffening mechanism provides a direct explanation for the milder nonlinear response obtained in the cantilever case.

The total simulation time was 1 day 22 h 21 min, with the mesh partitioned into four parts. In both cases, the harmonic balance method was applied with the same number of harmonics, ensuring a consistent basis for comparison. The overall amplitude of the response is in agreement with the results reported by Blahoš, although a slight frequency shift is observed. This discrepancy can be attributed to implementation details: in Blahoš's study, the excitation force was distributed over four nodes at the beam tip, whereas in the present work it is applied to a single node.

Parameter	**Value**
Newton tolerance	5×10^{-4}
Maximum Newton steps	15
Initial continuation step	8×10^{-2}
Minimum continuation step	10^{-6}
Maximum continuation step	1
Step increase factor	1.1
Step decrease factor	0.4
System harmonics	H_{012345}
Number of unknowns	$227,205$
Excitation force amplitude	2 N
Predictor type	Tangent
Corrector type	Pseudo-arclength

Table 5.2.7: Simulation parameters used for the NLFR computation of the cantilever beam.

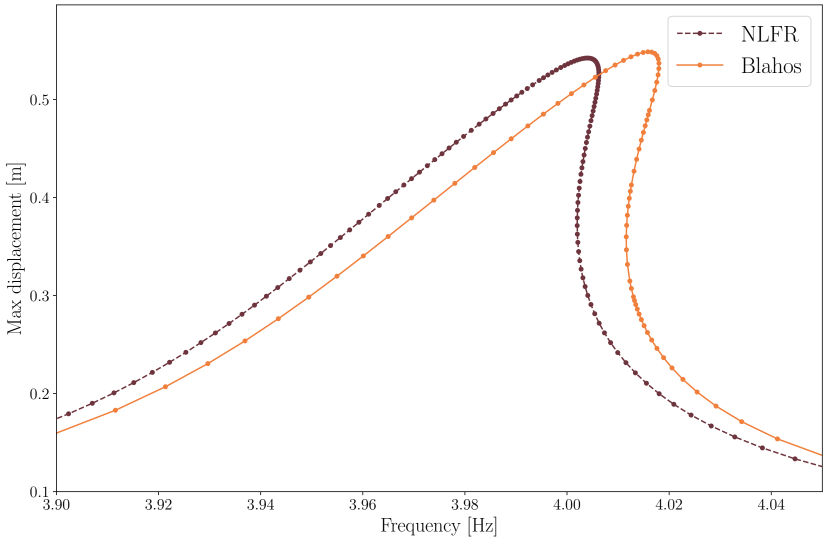


Figure 5.2.7: NLFRs of the cantilever beam. The results are compared with those reported in the doctoral thesis of Blahoš [13].

5.2.2.3 Nonlinear Frequency Response vs Backbone Curve

The NNM is represented by the backbone curve shown in Figure 5.2.8, using the continuation parameters listed in Table 5.2.8. The total computation time was 7 hours and 20 min. The backbone curve and the peak amplitudes of the NLFRs align closely, confirming the consistency between the forced and unforced nonlinear responses.

Parameter	**Value**
Newton tolerance	10^{-4}
Maximum Newton steps	20
Initial continuation step	10^{-4}
Minimum continuation step	10^{-6}
Maximum continuation step	1.1
Step increase factor	1.2
Step decrease factor	0.4
System harmonics	H_{012345}

Table 5.2.8: Numerical parameters used for the NNM computation of the cantilever.

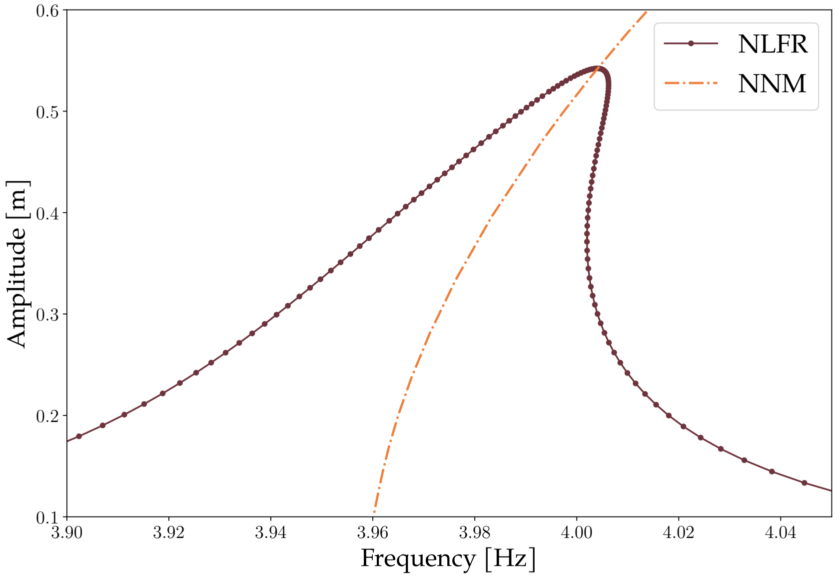


Figure 5.2.8: Comparison between the backbone curve and the resonance peak amplitudes of the NLFRs for the cantilever.

After assessing both the cantilever and clamped–clamped configurations through NNM and NLFR analyses, it was observed that the chosen step-size lacked robustness along the frequency response curves. This issue motivates a dedicated investigation of continuation strategies, presented in Section 5.3.

5.3 Comparison of Continuation Strategies

This section presents a comparative analysis of the continuation strategies employed in this work. Two distinct aspects are considered: the influence of the step-size control mechanism, discussed in Section 5.3.1, and the choice of predictor-corrector formulation, examined in Section 5.3.2. Both aspects play an essential role in determining the robustness, efficiency, and computational cost of the continuation process.

5.3.1 Comparison of Step-Size Methods

The step-size strategy is a key component of continuation, as introduced in Section 3.4.5. To evaluate its impact, a comparison is carried out between the iteration-based and angle-based refinement strategies. The test case corresponds to the clamped–clamped beam of Figure 5.2.1, discretised with a $4 \times 4 \times 10$ HEX-P1 mesh. Retaining the harmonic H_{012} results in 275 nodes and $25,515$ unknowns, enabling an accurate assessment of the continuation procedure.

Both strategies employ identical predictor and corrector schemes, so that differences arise solely from step-size control. The corresponding parameter sets are given in Table 5.3.1. The iteration-based strategy is intentionally not optimised, to highlight its typical limitations, whereas the angle-based formulation is sensitive to its tuning parameters, as discussed in Section 3.4.5.2.

Parameter	**Iteration-Based Strategy**	**Angle-Based Strategy**
Minimum step (s_{min})	10^{-6}	10^{-6}
Maximum step (s_{max})	1	5
Initial step (s_{start})	1	1
Maximum Newton iterations (N_{max})	10	8
Step increase factor (ξ_{up})	1.1	–
Step decrease factor (ξ_{down})	0.1	0.05
Optimal angle (γ_{opt})	–	0.05
Small-angle exponent (ψ_{in})	–	10^3
Large-angle exponent (ψ_{out})	–	10^2

Table 5.3.1: Parameter sets for the step-size strategies.

The results are shown in Figure 5.3.1, and the numerical compariso n appears in Table 5.3.2. The iteration-based method does not consider the curvature of the solution path and therefore limits the growth of the step size, slowing down convergence. The angle-based strategy, on the other hand, adapts the step size based on the local geometry of the curve: it increases the step when the path is nearly linear and reduces it when the curvature rises. This behaviour allows larger steps in smooth regions while maintaining accuracy near turning points, resulting in improved overall performance.

Strategy	**Points**	**Newton Iterations**	**Computation Time**
Iteration-based	232	$1,200$	3 h 11 min
Angle-based	134	755	2 h 16 min

Table 5.3.2: Comparison of continuation strategies applied to a clamped–clamped beam.

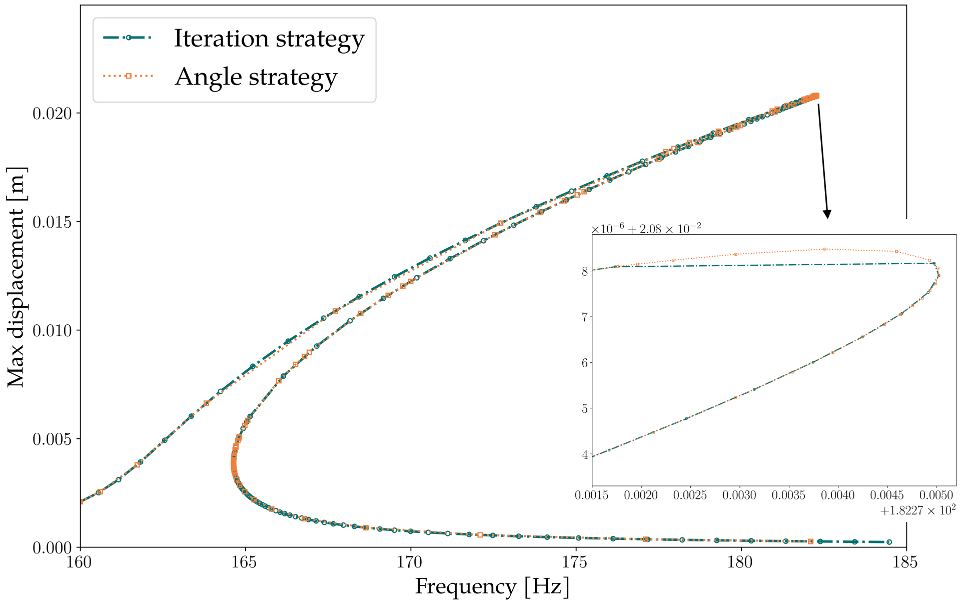


Figure 5.3.1: Evolution of the continuation step size for iteration-based and angle-based methods.

5.3.2 Comparison of Predictor–Corrector Methods

The second part of the analysis concerns the influence of predictor–corrector schemes. Several combinations introduced in Section 3.4.4 were tested on the same clamped–clamped beam under NLFR conditions, and the results are summarised in Table 5.3.3.

The secant predictor combined with a pseudo-arclength corrector yields the best compromise between computational cost and robustness. This approach requires the fewest continuation points and Newton iterations, leading to the shortest total runtime. The secant predictor is efficient when the path is sufficiently smooth, reducing the burden on the corrector phase.

The tangent predictor, although generally more robust near turning points, is less efficient in this configuration, since $\nabla_\omega \boldsymbol{G}$ was approximated by finite differences, which may degrade prediction quality near resonances. The spherical arclength method introduces additional overhead due to its tighter constraint formulation, reducing efficiency.

These observations agree with previous studies, such as Kadapa [94] on extrapolated predictors and Dickson et al. [95] on conditioning issues near turning points.

Predictor	Corrector	Points	Newton Iter.	Time
Tangent	Pseudo-Arclength	208	684	1 h 15 min
Tangent	Arclength	237	780	1 h 27 min
Secant	Pseudo-Arclength	172	567	59 min

Table 5.3.3: Comparison of different predictor-corrector method combinations applied to a clamped–clamped beam.

5.4 Validation of the Rotating-Frame Formulation

To verify the correctness of the rotating-frame formulation introduced previously, four complementary validation steps are carried out. First, the axial deformation induced by rotation is analysed by comparing the numerical results with the corresponding analytical solution, as detailed in Section 5.4.1. Second, the accuracy of the computed Campbell diagram is assessed to ensure that the formulation correctly captures the rotation-induced frequency shifts, including the contribution of geometric nonlinear effects, as discussed in Section 5.4.2.

Next, the ability of the proposed formulation to accurately reproduce nonlinear modal characteristics is examined through a verification of the computed NNMs, as presented in Section 5.4.3. Finally, the results obtained from the NNMs analysis are extended to the study of the forced dynamics by means of a NLFRs analysis, which is described in Section 5.4.4.

5.4.1 Verification Through Axial Deformation

A first validation is carried out by comparing the numerical axial deformation with the analytical solution available for a rotating cantilever in static conditions. The benchmark configuration follows the setup of Martin et al. [67]. The cantilever has dimensions $1 \times 0.1 \times 0.005$ m and rotates about the global z-axis with an offset $R = 0.1$ m. The finite-element model used in this work employs a $2 \times 30 \times 8$ HEX-P2 mesh. The geometry is shown in Figure 5.4.1, and the material properties are listed in Table 5.4.1.

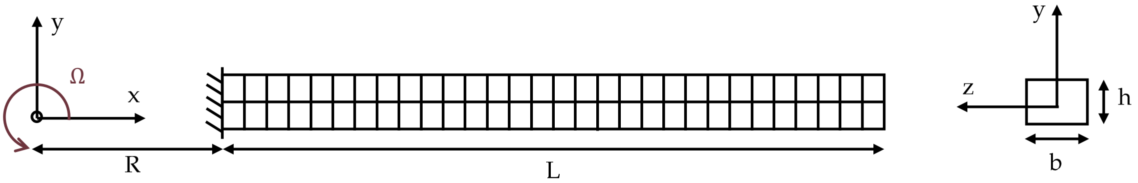


Figure 5.4.1: Rotating cantilever beam used for validating the rotating-frame formulation [67].

Parameter	Unit	Value
Density	$[\mathrm{kg\,m^{-3}}]$	4,400
Young's modulus	$[\mathrm{N\,m^{-2}}]$	1.04×10^{11}
Poisson's ratio	[–]	0.3

Table 5.4.1: Material properties of the rotating cantilever.

The analytical solution for the axial displacement U_x is derived by considering only the linear term, as assumed in the analytical formulation [96]. The governing differential equation is

$$\frac{\partial^2 U_x}{\partial x^2} + \frac{\rho}{E}\varpi^2 x = 0, \qquad U_x(0) = 0, \quad \frac{\partial U_x}{\partial x}(L) = 0. \tag{5.4.1}$$

Integration gives

$$U_x(x) = \frac{\rho\varpi^2}{2E}\left(xL^2 - \frac{x^3}{3}\right). \tag{5.4.2}$$

The comparison in Figure 5.4.2 demonstrates the excellent agreement between the analytical and numerical predictions at $\varpi = 500$ RPM, confirming the correct implementation of the centrifugal stiffening effects in the rotating-frame formulation.

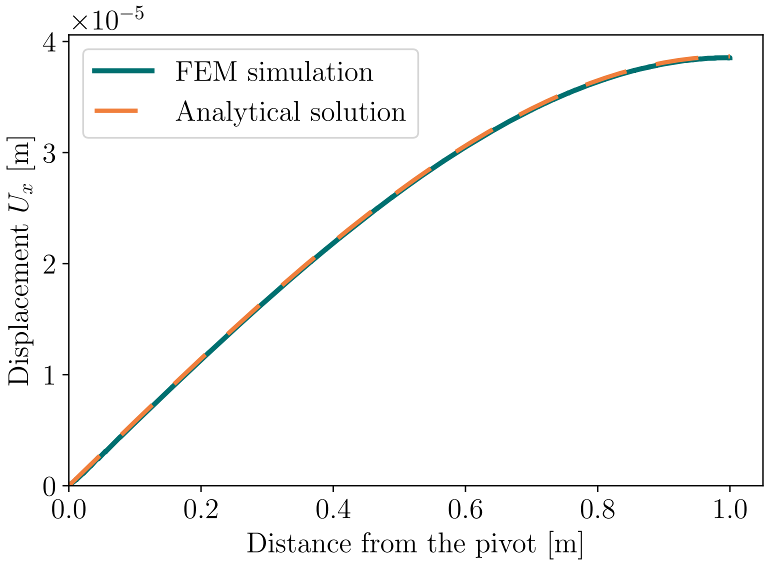


Figure 5.4.2: Comparison between the analytical [96] and numerical axial displacements.

5.4.2 Verification Through Campbell Diagram

A second validation concerns the ability of the rotating-frame formulation to reproduce the evolution of natural frequencies with rotation speed, classically represented in a Campbell diagram. The benchmark follows the rotating cantilever considered in [67], and the numerical frequencies obtained in this work are compared with the reference values reported in the literature. The corresponding results are summarised in Table 5.4.2.

The Campbell diagram computed numerically is presented in Figure 5.4.3. As expected, the flap-wise bending frequencies increase with the rotational speed due to centrifugal stiffening, whereas the edge-wise mode exhibits a more moderate evolution because of its higher bending rigidity. The torsional frequency remains nearly constant over the full rotation range.

The computation times for this example were as follows: the static analysis required 7 s, the eigenfrequency computation required 32 s, and the model contained 15 discretisation points and 15,555 DOFs, yielding a total runtime of 39 s.

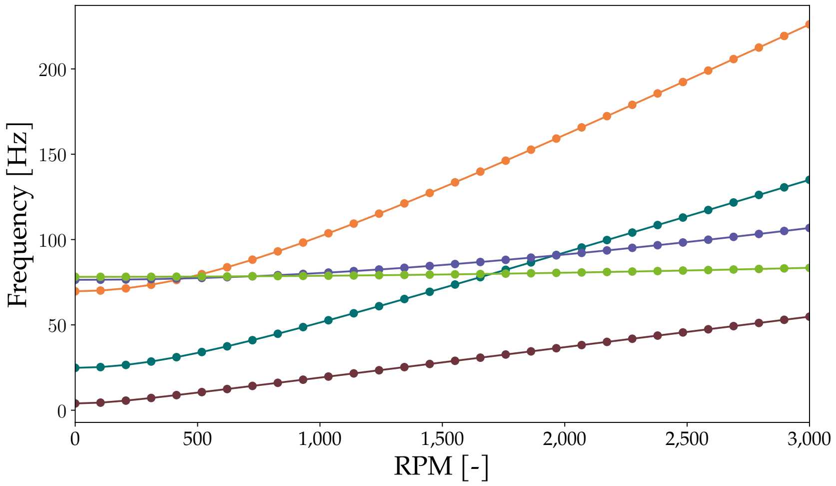


Figure 5.4.3: Campbell diagram of the rotating cantilever. Colour code: [——] 1F, [——] 2F, [——] 3F, [——] 1E, [——] 1T.

Table 5.4.2 shows an excellent agreement between the present results and the reference data. The relative error is nearly zero for all the considered operating conditions. This confirms the accuracy of the rotating-frame formulation in capturing the centrifugal stiffening effect and the associated frequency shift.[1]

RPM $[-]$	0	500	$1,000$	$2,000$	$3,000$
f_{1F} [Hz] [67]	3.961	10.322	19.114	36.950	54.826
f_{1F} [Hz]	3.961	10.321	19.114	36.950	54.826
ε_1 [%]	0.00	0.01	0.00	0.00	0.00
f_{2F} [Hz] [67]	24.837	33.575	51.371	92.425	135.021
f_{2F} [Hz]	24.837	33.575	51.371	92.425	135.021
ε_2 [%]	0.00	0.00	0.00	0.00	0.00

Table 5.4.2: Comparison between Campbell diagram frequencies reported in [67] and those computed in this work.

5.4.3 Nonlinear Normal Mode Verification

The cantilever beam is replaced by an alternative configuration with a length $L = 1$ m, a thickness $h = 0.02$ m, a width $b = 0.03$ m, and a root fillet radius $R = 0.1$ m, while preserving the material properties previously defined in Table 5.4.1. The structure is discretised using a $30 \times 2 \times 2$ HEX-P2 finite element mesh. The nonlinear backbone curves are computed using a harmonic balance formulation with 5 harmonics, leading to a total of 50,325 unknowns. A comparison is performed with the nonlinear normal mode results reported by Martin et al. [67], as illustrated in Figure 5.4.4. A slight lack of smoothness can be observed in the computed backbone curves, suggesting that the time step employed in the present simulations may be marginally too large. Nevertheless, a good overall agreement is obtained between the two sets of results across the full range of rotational speeds. The discrepancy observed at 0 RPM is consistent with that found at higher rotational speeds, indicating that the difference does not originate from the formulation itself but rather from differences in the adopted finite element discretisation, which exhibits a lower effective stiffness. In addition, the backbone curves clearly exhibit a transition from a hardening to a softening nonlinear behaviour as the rotational speed increases. A summary of the backbone computation times for the different rotational speeds is provided in Table 5.4.3.

---

[1] The numerical results are reported with a precision limited to three digits after the decimal point.

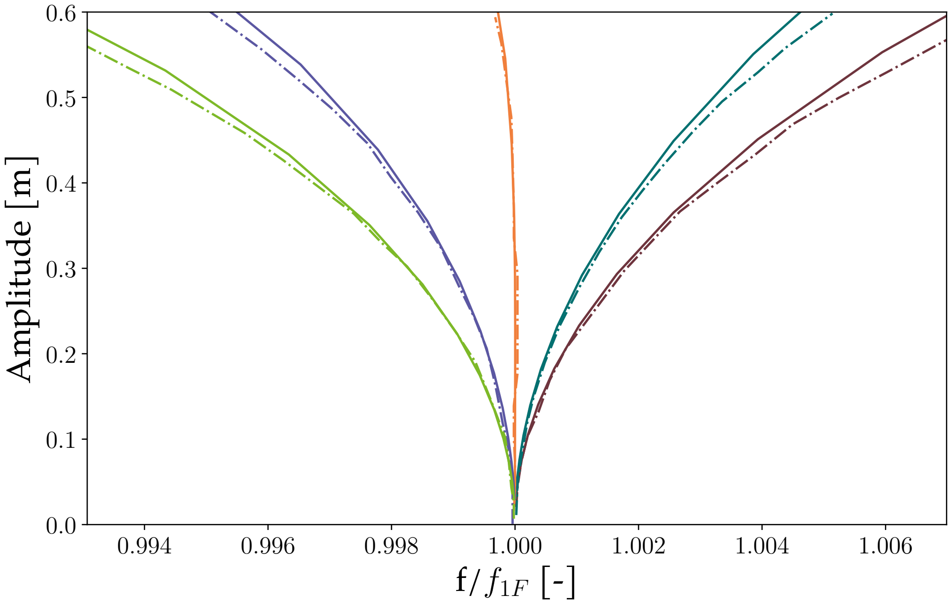


Figure 5.4.4: Backbone curve associated with mode 1F for different rotational speeds. Continuous lines [−] correspond to results obtained using Quanscient, while dash-dotted lines [−·] represent the results reported by Martin et al. [67]: [—] 0 RPM, [—] 500 RPM, [—] 1,000 RPM, [—] 1,500 RPM, and [—] 2,000 RPM.

RPM	Iter New.	Nbr Points	Time
0	51	13	00:31:04
500	50	13	00:33:17
1,000	37	12	00:26:16
1,500	44	12	00:22:31
2,000	48	13	00:31:15

Table 5.4.3: Summary of backbone computations for different rotational speeds performed using 5 harmonics and a total of 50,325 unknowns. All computation times are reported in the compact hh:mm:ss format.

5.4.4 Nonlinear Frequency Response Analysis

Based on the consistency observed in the nonlinear normal modes, the analysis is extended to NLFR functions. Due to the centrifugal stiffening induced by rotation, the dynamic characteristics of the system evolve with the rotational speed. In order to preserve a comparable level of vibration amplitude in the nonlinear frequency response analyses, the forcing amplitude is increased as the rotational speed increases. In parallel, the damping properties are adjusted to ensure a consistent level of energy dissipation. Specifically, the damping ratio is increased such that it remains equal to 0.05% for all considered rotational speeds. –The system is excited by a harmonic force defined as the cosine of the first harmonic, applied at the free end of the cantilever at a single point located at the tip, and acting along the x-direction. TheNLFR simulations are performed using the same number of harmonics as those retained for the nonlinear normal mode computations, in order to ensure full consistency between the two analyses.

The NLFRs and their associated backbone curves obtained for different rotational speeds are presented in Figure 5.4.5. A consistent qualitative agreement is observed between the NLFRs and the corresponding NNMs, indicating that the dominant nonlinear dynamics are consistently captured by the model across both types of analysis. A summary of the NLFR simulation computation times is provided in Table 5.4.4.

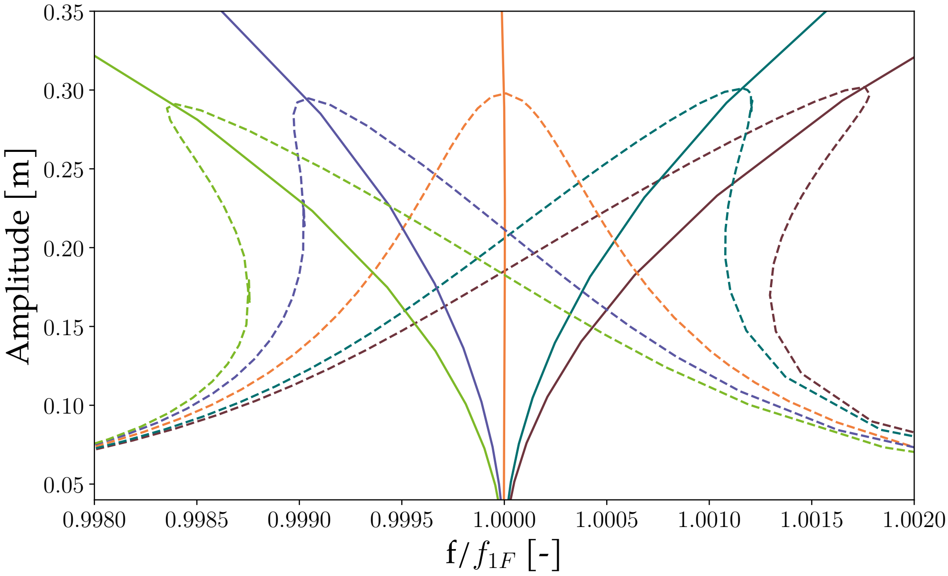


Figure 5.4.5: NLFRs for the mode 1F and associated backbone curves for different rotational speeds: [——] 0 RPM, [——] 500 RPM, [——] 1,000 RPM, [——] 1,500 RPM, and [——] 2,000 RPM.

RPM	Force [N]	Iter New.	Nbr Points	Time
0	2	167	41	1:23:12
500	2.21	222	37	1:15:19
1,000	2.79	345	39	1:15:52
1,500	3.73	323	38	1:09:44
2,000	4.94	304	44	1:04:26

Table 5.4.4: Summary of NLFRs computations for different rotational speeds performed using 5 harmonics and a total of 50,325 DOFs. All computation times are reported in the compact hh:mm:ss format.

5.5 Scalability

Scalability is a key performance indicator for any parallel solver, measuring its capacity to efficiently exploit increased computational resources. In this work, parallelisation is handled through the MPI, where each MPI process runs independently and communicates with others to share the computational load. The number of MPI processes thus reflects the level of available parallel computing power.

Two main notions of scalability are commonly defined [13]. Strong scalability evaluates how the total runtime decreases as the number of MPI processes increases for a fixed overall problem size. Weak scalability, on the other hand, examines whether the runtime remains constant when both the number of processes and the problem size increase proportionally, maintaining a constant workload per MPI process. In the present work, only strong scalability is investigated, as the weak scalability analysis cannot be carried out directly due to the lack of control over the number of elements assigned to each mesh partition in QUANSCIENT ALLSOLVE.

In this study, a single computational thread is assigned to each MPI process, such that the number of MPI processes corresponds directly to the number of mesh partitions. Although the raw runtime data provided by Quanscient includes various stages such as initialisation, output writing, or mesh generation, these steps do not scale meaningfully with the number of processes. As such, they do not provide useful insights into the parallel performance. The analysis therefore focuses on the most representative tasks: the Jacobian setup and the solve phases using a direct solver MUMPS.

The Jacobian setup refers to the repeated construction of the system matrix and the right-hand side vector at each nonlinear iteration. The solve phase corresponds to the execution of the linear solver, which includes the matrix factorisation of the system. Both phases are central to the numerical performance of the code.

The following subsections are organised as follows. First, the common numerical settings and model specifications used in all scalability tests are presented in Section 5.5.1. Then, the performance is assessed on a reduced configuration in Section 5.5.2, where additional aspects such as quadruple precision and the comparison between AFT and analytical solutions are also addressed. Finally, a larger testcase is considered in Section 5.5.3, providing insights into the solver's behaviour under increased computational demands.

5.5.1 Reference configuration for scalability tests

All scalability analyses presented in this work are based on the clamped–clamped beam described in Figure 5.2.1. While the physical model remains identical across all tests, the mesh density, harmonic content, and specific computational strategies are varied depending on the scalability scenario under consideration. The corresponding parameters for each case are detailed in the relevant sections and summarised in the associated tables.

For all scalability tests, the solver converged in four Newton iterations at an excitation frequency of 155 Hz, with a residual tolerance of 5×10^{-6}. These convergence settings are consistent throughout the study to ensure that performance differences arise solely from the parallelisation and computational configurations, rather than from changes in numerical tolerance or stopping criteria.

5.5.2 Strong Scalability Small Testcase

Two configurations were considered to obtain a similar number of unknowns while addressing different aspects of the problem: one examining the impact of the number of nodes, and the other assessing the impact of the number of harmonics on the system. These configurations are summarised in Table 5.5.1.

For each test case, Figure 5.5.1 reports both the total runtime and its distribution among the main computational tasks, together with the corresponding parallel efficiency. The results associated with test 1 are shown on the left, while those corresponding to test 2 are presented on the right.

Parameter	**Test 1**	**Test 2**
Mesh	$2 \times 2 \times 512$ HEX-P1	$2 \times 2 \times 188$ HEX-P1
Total nodes	4,617	1,701
Harmonics	H_{0123}	$H_{0123456789}$
Number of unknowns	538,125	537,225

Table 5.5.1: Simulation configurations for the scalability test using the clamped–clamped beam.

The runtime breakdown highlights the relative contributions of the linear system solution, the Jacobian and right-hand side assembly, and the remaining algorithmic operations. This analysis is performed analytically, without relying on the AFT; a comparison with results obtained using the AFT will be presented in the following sections.

The results show that Test 2 requires nearly three times more computation time than Test 1 for almost all MPI configurations. This is mainly due to the fact that Test 2 involves three times more harmonics for practically the same number of unknown, which leads to a denser system matrix with reduced sparsity, thereby increasing the computational burden.

The overall scalability is ultimately limited by the interconnectivity of the system, which constrains parallel performance as the number of MPI processes increases. This effect becomes more pronounced when the harmonic content is high, as the MPI processes require more frequent and larger data exchanges, leading to a faster degradation of the Jacobian assembly performance.

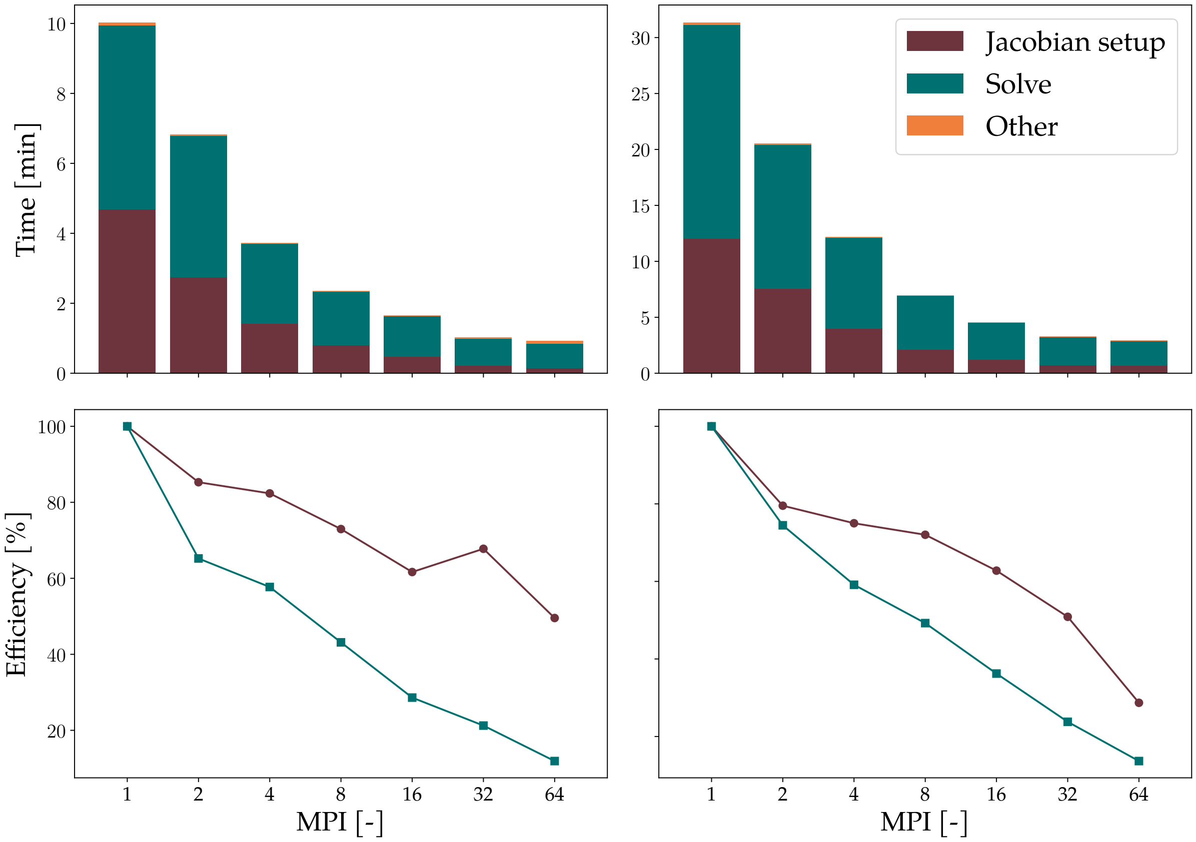


Figure 5.5.1: Scalability analysis for Test 1 (left) and Test 2 (right), with the Jacobian evaluated analytically.

5.5.2.1 Alternating Frequency-Time and Analytical Formulations

The same setup described in Table 5.5.2 is used here, but employing the AFT method. The number of time samples is systematically set to $N = 2(2n_H + 1)$, and the scalability of this approach is compared against the analytical formulation.

As shown in Figure 5.5.2, the analytical formulation performs better for Test 1 (left) when only a small number of harmonics is considered. In contrast, for Test 2 (right), the opposite trend is observed as the number of harmonics increases. To gain further insight into this behaviour, a more detailed analysis of the solver and of the problem generation process is subsequently carried out.

It can be observed that the solve phase always requires less time when using the AFT formulation. This is because the coefficient matrix produced by AFT is structurally sparser than the one obtained with the fully analytical formulation. In the analytical case, all harmonic couplings are retained without approximation, which results in a denser matrix due to strong inter-harmonic interactions. In contrast, the AFT approach includes all harmonic coefficients but applies a threshold, setting to zero those whose magnitude is below a given limit. This increases sparsity and reduces the computational cost. When this threshold is removed, the AFT matrix becomes as dense as the analytical one, and the solver time becomes comparable between the two approaches.

When constructing the Jacobian of the residual, a crossover with respect to the harmonic order n_H can be observed. For small values of n_H, the AFT procedure is slower due to the fixed overhead associated with FFT and sampling operations. As n_H increases, however, the analytical Jacobian becomes progressively more expensive. This is a direct consequence of the geometrically exact kinematics, where the Green–Lagrange strain introduces nonlinear terms up to the cubic order in the displacement field. These contributions generate cross-harmonic couplings that scale as $\mathcal{O}(n_H^3)$, thereby leading to a rapid growth of the computational cost. In contrast, the AFT approach avoids assembling these dense couplings and retrieves the Jacobian information through FFT/IFFT operations at approximately $\mathcal{O}(n_H \log n_H)$, so that beyond moderate values of n_H its setup becomes significantly faster.

In parallel, AFT scales better for the Jacobian setup. Most operations are local (FFT and pointwise evaluations), so only small border-data exchanges (halo exchanges) are needed. The analytical setup creates many cross-harmonic links, which makes these exchanges larger and adds more global synchronisations. As the number of MPI processes increases, AFT therefore benefits more from parallelism and maintains higher strong-scaling efficiency.

Nevertheless, at high MPI counts, both approaches eventually reach a saturation point, where communication and synchronisation overhead dominate. As a result, their overall runtimes become nearly identical regardless of the technique use.

This testcase can be directly compared to the one presented by Blahoš in his doctoral thesis [13], where Test 1 is almost identical, except that his case involves 323,001 unknowns while ours contains fewer, with the same number of harmonics. In Blahoš' results, the computation time was 35 min for 1 MPI process and 42 s for 64 MPI processes. These computations were executed on a supercomputing machine such as Salomon, which offers higher performance than AWS instances. It can therefore be observed that QUANSCIENT ALLSOLVE outperforms the results reported in the thesis, despite involving more unknowns and running on non-supercomputing hardware. Nevertheless, when considering the parallel efficiency, Blahoš' results remain consistently above 80% up to the highest MPI count tested.

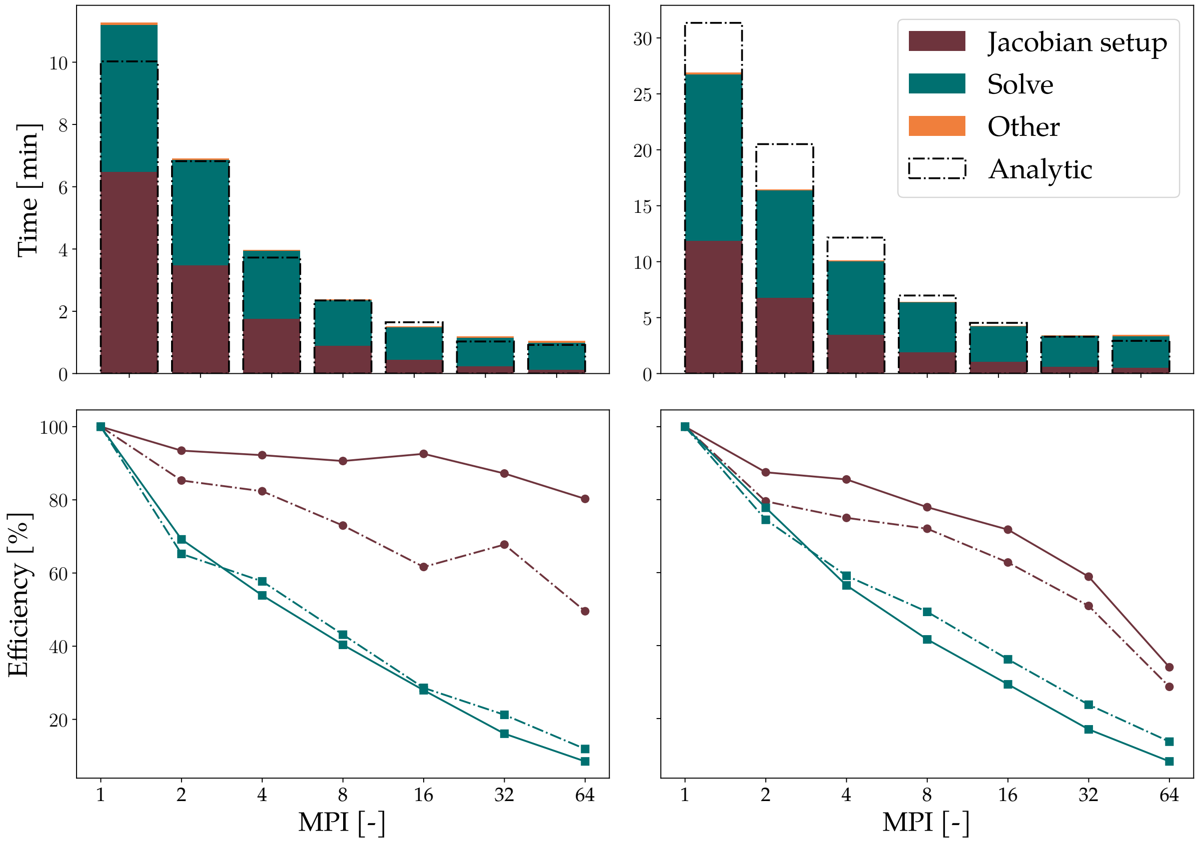


Figure 5.5.2: Scalability analysis for Test 1 (left) and Test 2 (right). The dashed line (-.) represents the analytical formulation, while the solid line (–) corresponds to the AFT approach.

5.5.2.2 Scalability of the Solver in Quadruple Precision

The quadruple-precision solver, based on the IEEE 754 `float128` standard, was evaluated for AFT analysis in Test 2 (see Table 5.5.2). The objectives are twofold: first, to quantify the additional computational cost introduced by quadruple precision compared to double precision; and second, to assess whether the solver maintains satisfactory scalability as the number of MPI processes increases. High-precision computations are particularly relevant when the system operates close to resonance, where the algebraic system becomes severely ill-conditioned and the solution is highly sensitive to rounding errors, or when solving large-scale multiphysics problems where accumulated floating-point errors may significantly degrade the accuracy.

As shown in Figure 5.5.3, the quadruple-precision analysis is approximately $7.4\times$ slower than double precision for 1 MPI process. This slowdown decreases progressively with the number of MPI processes, reaching about $3.06\times$ at 64 MPI. There is no universal slowdown factor: in this case, the ratio varies between $3\times$ and $8\times$, but it consistently remains significantly higher than in double precision.

When examining the runtime distribution, it is observed that the Jacobian setup phase becomes the dominant contributor in quadruple precision, whereas in double precision the solve phase was dominant. The focus here is on strong scaling efficiency, also reported in Figure 5.5.3. The Jacobian setup scales efficiently up to 4 MPI processes, after which its efficiency drops sharply. In contrast, the solve phase maintains the highest efficiency across all MPI counts, even though it no longer dominates the total runtime.

The performance gap between quadruple and double precision is primarily explained by hardware limitations. On modern `x86_64` cpu, `float128` operations are not supported natively; instead, they are implemented in software (e.g. via `libquadmath`). This implementation relies on decomposing each quadruple-precision operation into a sequence of multiple `float64` operations, which are natively supported and hardware-accelerated. Such software emulation introduces a substantial constant cost per operation and doubles the memory footprint, thereby increasing memory bandwidth pressure. These architectural constraints directly explain both the observed slowdown and the shift in runtime profile compared to double precision. Nevertheless, the detailed efficiency behaviour cannot be fully interpreted, since the lack of access to the underlying source code prevents a precise understanding of whether it originates from algorithmic choices, emulation strategies, or low-level implementation details.

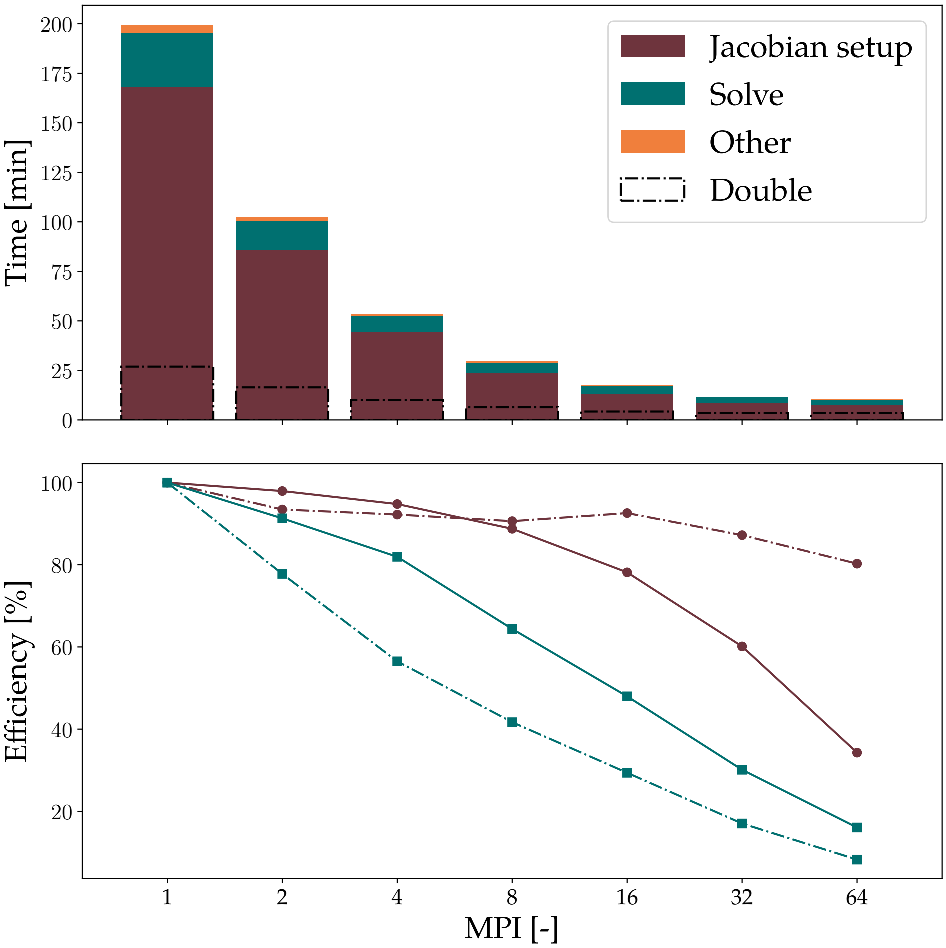


Figure 5.5.3: Scalability results for Test 2 with the AFT analysis. The figure reports the total code runtime and the parallel efficiency for double-precision (-.) and quadruple-precision (—) computations.

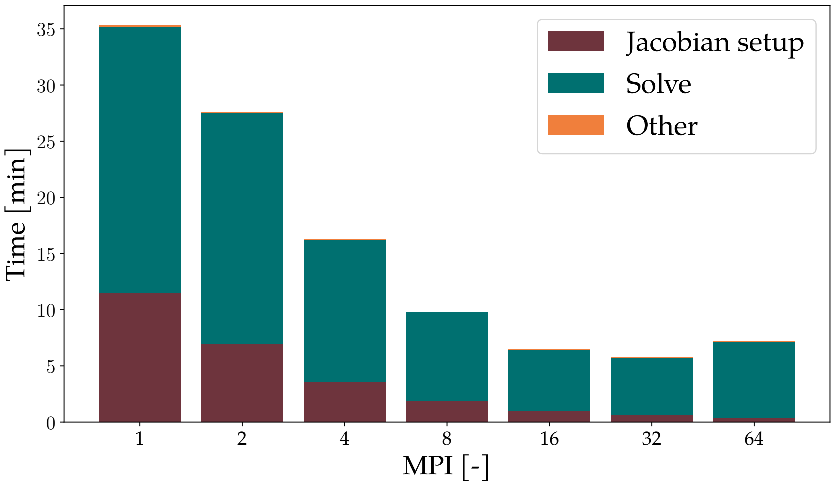
5.5.3 Strong Scalability Large Testcase

The following analysis repeats the same strong scalability test as presented for the smaller case, but now with a finer mesh to create a substantially larger problem size. This approach allows the investigation of whether the scaling behaviour observed previously remains valid when the number of degrees of freedom is significantly increased. The simulation parameters for this larger testcase are summarised in Table 5.5.2. This test is based on the analytic simulation.

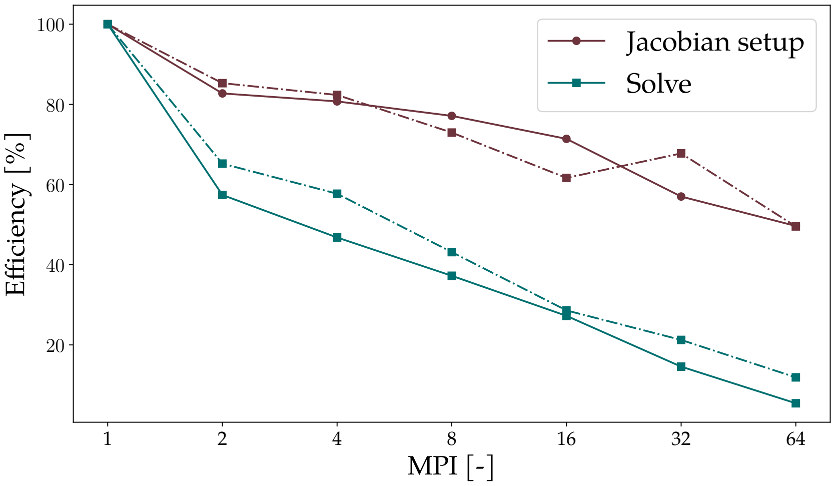
Parameter	**Value**
Mesh	$4 \times 4 \times 300$ HEXA-P1
Total nodes	$7,525$
Harmonics	H_{0123}
Number of unknowns	$1,022,301$

Table 5.5.2: Simulation configurations for the larger scalability test using the clamped–clamped beam.

As illustrated in Figure 5.5.4a, the optimal performance is reached with 32 MPI ranks. Beyond this point, increasing to 64 ranks results in a longer wall time rather than an improvement. A comparison with a smaller case (see Figure 5.5.3) reveals that such behaviour only appears for the larger system, whereas the scaling remains effective up to 64 ranks for the smaller one. To better understand this discrepancy, the efficiency is presented in Figure 5.5.4b. The Jacobian assembly exhibits nearly the same performance, but the solver scales significantly worse as the number of unknowns grows. This loss of efficiency can therefore be attributed to the MUMPS solver. Similar results are reported in the literature [97]. These findings indicate that the solver constitutes the main bottleneck. A deeper investigation, including tests with different mesh sizes, partitioning strategies, and possibly alternative solvers, would be required to fully characterise this behaviour. Such an extended analysis, however, falls beyond the scope of this thesis.



(a) Code runtime for the large scalability test.



(**b**) Parallel efficiency for the large test case (—) and the small test case 1 (–·).

Figure 5.5.4: Scalability analysis for the large test case, with the Jacobian evaluated analytically.

5.6 Industrial Case

After validating the methodology and assessing the scalability of Quanscient Allsolve, an industrial test case of high engineering relevance is considered on a fan blade geometry taken from [6]. The blade pre-twist induces significant quadratic nonlinearities, while its shell-like behaviour further increases the problem complexity. The mesh, shown in Figure 5.6.1, consists of $2,041$ TET-P2 elements and $3,895$ nodes, with the blade clamped at its root. The bounding box dimensions are $\Delta X = 0.41$ m, $\Delta Y = 1.14$ m, and $\Delta Z = 0.35$ m. The blade span is approximately 1.02 m, with a chord length of 0.48 m and a maximum mid-chord thickness of 0.01 m. A single nodal force is applied for the NLFR computation, as illustrated in the diagram and discussed later. The corresponding material properties are listed in Table 5.6.1.

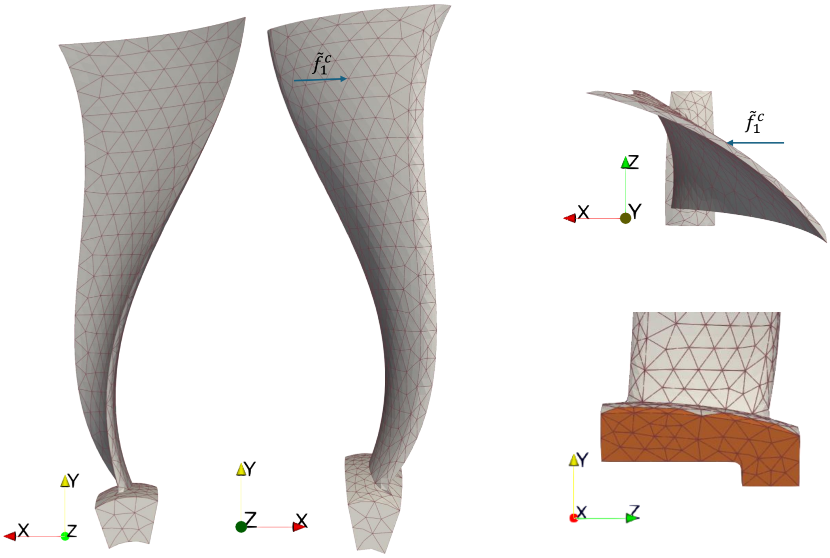


Figure 5.6.1: Fan blade test case showing the FEM mesh and boundary conditions. The blade is clamped at the root (orange), and a nodal force $\tilde{\boldsymbol{f}}_1^c$ is applied near the tip.

Parameter	**Unit**	**Value**
Density	$[\mathrm{kg\,m^{-3}}]$	$4,400$
Young's modulus	$[\mathrm{N\,m^{-2}}]$	1.04×10^{11}
Poisson's ratio	[–]	0.3
Damping matrix	[–]	$\boldsymbol{D} = 0.3 \times \boldsymbol{M}$

Table 5.6.1: Physical parameters for the fan blade.

5.6.1 Mode Shapes and Eigenfrequencies

The eigenfrequencies and corresponding modes are for the presented in Table 5.6.2 and Figure 5.6.2. The frequency ratio of the first mode is close to 3:1 with respect to the second and third modes, suggesting potential.

Mode	1F	2F	1T	1E
Frequency [Hz]	15	41.68	49.89	174.74
Ratio f_n/f_{1F} [-]	1	2.78	3.33	11.65

Table 5.6.2: Eigenfrequencies of the fan blade and their ratios with respect to the first Flap-wise bending mode frequency.

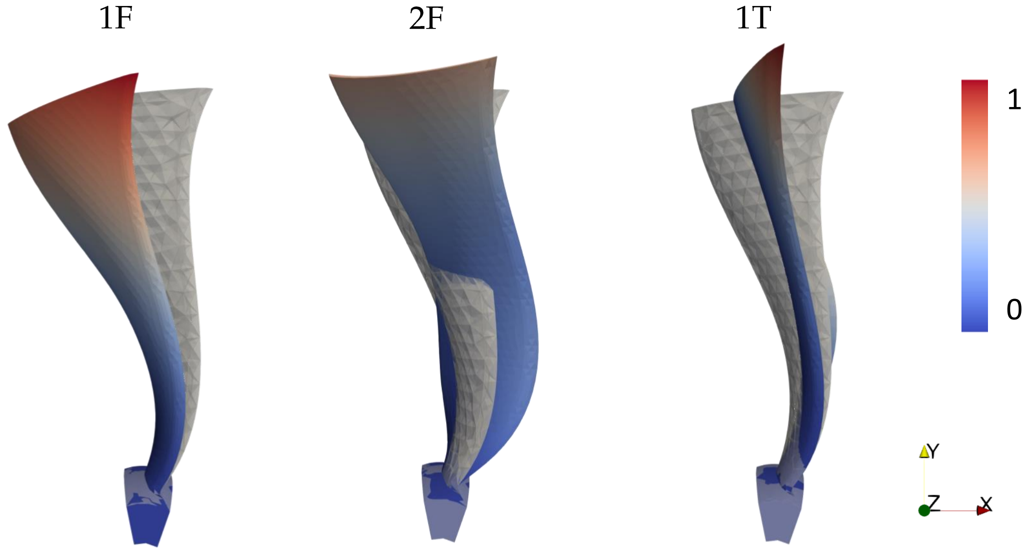


Figure 5.6.2: The third mode shape of the fan blade.

5.6.2 Nonlinear Frequency Response with Backbones

The NLFR targets the first eigenfrequency. A single nodal excitation force is applied to the cosine component of the first harmonic $\tilde{\boldsymbol{f}}_1^c$, aligned with the x-direction, and positioned just below the tip, as shown in Figure 5.6.1. This loading primarily excites the first two bending modes, while the third torsional mode remains unexcited at the considered amplitude. The continuation parameters are listed in Table 5.6.3.

Parameter	Value
Newton tolerance	5×10^{-4}
Maximum Newton steps	15
Initial continuation step	0.05
Minimum continuation step	10^{-6}
Maximum continuation step	1
Step increase factor	1.1
Step decrease factor	0.4
System harmonics	H_{0123}
Number of unknowns	$11,685$
Excitation amplitude	20 N
Predictor type	Tangent
Corrector type	Pseudo-arclength

Table 5.6.3: Continuation parameters used for the NLFR computation of the fan blade.

The results presented in Figure 5.6.3 show a clear softening behaviour, consistent with trends reported in the literature for pre-twisted blade-like structures [6, 13]. The NLFRs and the backbone curve align closely, confirming the accuracy of the computation. This softening originates from the geometric coupling induced by the blade's pre-twist and shell-type geometry. When the blade vibrates in its fundamental flexural mode, the tip deflection generates coupled torsional and membrane deformations. These effects reduce the effective bending stiffness, making the structure more compliant as the vibration amplitude increases. Consequently, the nonlinear resonance frequency shifts towards lower values with increasing amplitude.

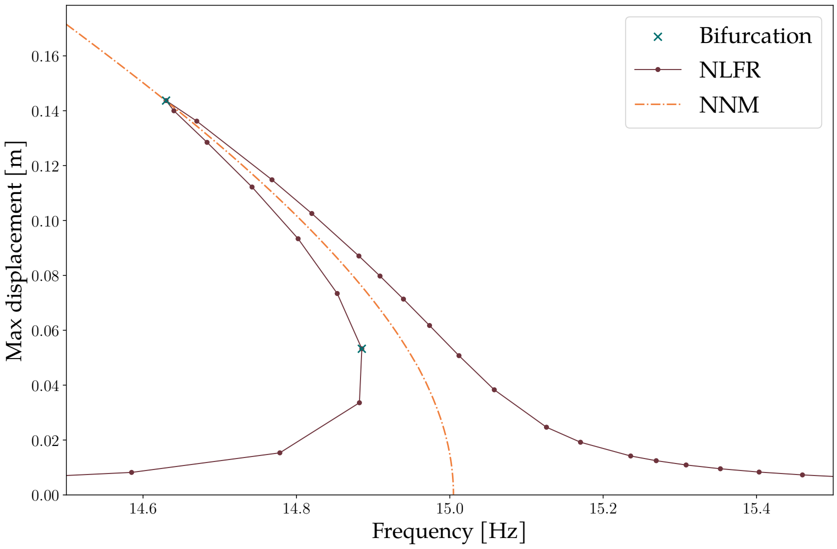


Figure 5.6.3: Backbone curve and the NLFRs of a fan blade.

5.6.3 Campbell Diagram

After the analysis performed at rest, the effect of rotation is now investigated. The blade is assumed to rotate about the z-axis, with its centre of rotation located at $(0, -0.37)\,\mathrm{m}$. The corresponding Campbell diagram is presented in Figure 5.6.4. In agreement with the cantilever results shown in Figure 5.4.3, the 1F increases with rotational speed as a consequence of the centrifugal stiffening effect. Unlike the simple beam case, for which the torsional mode remains nearly unaffected by rotation, the 1T of the blade exhibits a clear dependence on the rotational speed. Finally,2F shows a behaviour similar to that of the 1T, due to the coupling between flap-wise bending and torsion.

The numerical model contains 17 points and a total of 11,685 degrees of freedom. The static computation required 1 min 32 s, while the eigenfrequency analysis required 31 s.

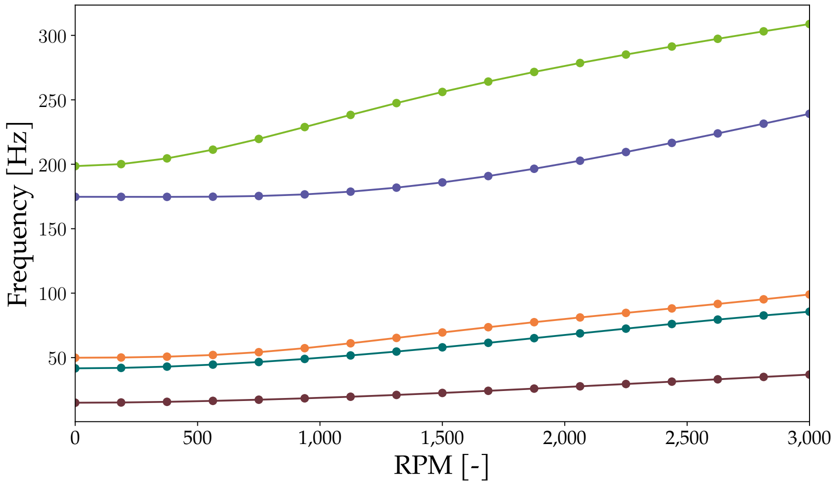


Figure 5.6.4: Campbell diagram of the fan blade. The colour code is as follows: [——] 1F, [——] 2F, [——] 1T, [——] 1E, [——] 3F.

5.7 Conclusion

The results obtained with Quanscient Allsolve confirm the successful integration of the harmonic balance continuation framework into a high-performance parallel computing environment. The first part of the study demonstrated that the methodology is able to reproduce, with a high degree of accuracy, the NLFRs and backbone curves of classical benchmark problems. These results were consistent with the reference solutions reported in the work of Blahoš, thereby providing a solid validation of the implementation and ensuring the credibility of the adopted formulation.

Beyond this verification step, an extensive investigation of numerical strategies was carried out. Different step-size control approaches were compared, and it was shown that the adaptive angle-based method provides the most reliable compromise between robustness and efficiency, although tuning of its parameters may be required and can be time-consuming. Similarly, the analysis of predictor–corrector schemes revealed that the secant predictor combined with a pseudo-arclength corrector consistently delivered the most accurate and stable results. By contrast, the tangent predictor, based on finite differences, introduced errors that limited its effectiveness in strongly nonlinear regimes. These observations establish a clear guideline for the selection of continuation strategies in future large-scale applications.

In addition, the rotating-frame formulation was rigorously verified through both analytical and numerical benchmarks. The analytical validation of the axial deformation confirmed the correct implementation of centrifugal stiffening effects, while the comparison with the cantilever benchmark reported by Touzé et al. for the nonlinear Campbell diagram and the computed NNMs demonstrated excellent agreement. These results were subsequently extended to the forced response through the computation of NLFRs, further confirming the robustness and consistency of the proposed formulation.

The scalability of Quanscient Allsolve was assessed through dedicated parallel computing tests. Strong scalability was demonstrated on both small and large test cases, with parallel efficiency maintained across increasing numbers of processes. A comparison between the AFT method and analytical formulations revealed that the AFT approach becomes preferable when a large number of harmonics is considered. The use of quadruple precision, compared to standard double precision, resulted in an eightfold increase in computational cost but confirmed the robustness of the framework under high-accuracy requirements. However, additional large-scale tests indicated that efficiency decreases due to the inability of the MUMPS solver to scale effectively for very large cases. This limitation indicates that further investigations should focus on alternative linear algebra solvers or improved preconditioning strategies to fully exploit the scalability potential of the platform.

Finally, the industrial case study provided a decisive demonstration of the practical relevance of the developed framework. The blade analysis confirmed that the methodology is capable of accurately capturing the NLFRs and the associated backbone curve in a realistic engineering configuration. This result shows that the approach is not limited to academic benchmarks but can be reliably applied to industrial-scale problems, thereby reinforcing its potential for practical deployment. In addition, the Campbell diagram was investigated, revealing interactions between modes through frequency crossings and veering phenomena, which highlight the influence of rotational speed on the dynamic behaviour and the onset of modal coupling.

All the results obtained from the NLFRs and NNMs computations are summarised in Table 5.7.1.

Structure	Unk.	Method	Node part.	Nbr Point	Iter New	Time
Clamped–clamped beam	12,375	NLFR	1	137	1,232	0:41:07
		NNM	1	56	218	0:27:32
Cantilever beam	227,205	NLFR	4	149	1,467	46:21:25
		NNM	1	24	309	7:19:33
Cantilever beam in rotation	50,325	NLFR	1	≈ 272	≈ 40	≈ 1:13:42
		NNM	1	≈ 46	≈ 13	≈ 00:28:53
Fan blade	81,795	NLFR	1	29	260	2:20:06
		NNM	1	9	41	0:24:45

Table 5.7.1: Summary of NLFR and NNM computations for the considered structures. For the rotating configurations, the reported values correspond to averages over the considered rotational speeds. All times are reported in the compact hh:mm:ss format.

This page intentionally left blank.

6

GENERAL CONCLUSION AND PERSPECTIVES

The present thesis was motivated by the ambition to bridge the gap between advanced nonlinear frequency-domain methods and their effective use in large-scale engineering applications. The central objective was to implement and validate a harmonic balance continuation framework within a high-performance computing environment, thereby enabling accurate and efficient nonlinear vibration analysis beyond academic benchmarks and into industrially relevant cases. This concluding chapter first synthesises the main achievements in Section 6.1, then critically examines the study's limitations in Section 6.2, and finally delineates perspectives and future work in Section 6.3.

6.1 Summary of Achievements

The work began with a comprehensive literature review on nonlinear frequency-domain analysis, continuation techniques, and solver technologies, which established the theoretical background and clarified the open challenges to be addressed (Chapter 2). Guided by these insights, a coherent methodology was constructed, ranging from the finite element formulation to the harmonic balance residual, together with a modular continuation framework encompassing predictor–corrector schemes and adaptive step-size control (Chapter 3). This formulation was further extended to account for computations in a rotating reference frame.

The approach was first exercised within PYHARM, a harmonic balance solver, which was extended in this thesis to compute conservative nonlinear normal modes through a dedicated phase condition and an augmented continuation scheme. This stage proved decisive for validating the formulation, identifying numerical pitfalls, and debugging developments in a controlled environment (Chapter 4).

Building on this foundation, the framework was then deployed in a high-performance setting via QUANSCIENT ALLSOLVE, enabling large-scale studies and systematic assessments on realistic finite element models (Chapter 5). Validation against reference solutions reported by Blahos confirmed the accuracy of the implementation for both NLFRs and backbone curves. In addition, the rotating-frame formulation was validated through comparisons with reference results reported by Touzé *et al.*, thereby confirming its consistency for the computation of Campbell diagrams, NNMs and NLFR.

Subsequent investigations compared numerical strategies, showing that an angle-based step-size adaptation offers a robust efficiency–stability compromise, albeit with parameters requiring careful tuning, while a secant predictor combined with a pseudo-arclength corrector delivered the most reliable convergence across strongly nonlinear regimes.

Scalability was evaluated through dedicated parallel tests. Strong scaling was maintained as the number of processes increased. The AFT formulation proved more effective when the number of harmonics grew. The use of quadruple precision, although incurring an approximately eightfold computational cost, confirmed robustness under stringent accuracy requirements. At larger scales, the observed loss of efficiency was traced to the linear-algebra backend (MUMPS), suggesting that alternative solvers and improved preconditioning strategies could further enhance the platform's scalability.

Finally, an industrial blade case demonstrated the practical relevance of the proposed framework, with the nonlinear frequency response and the associated backbone curve being accurately captured. These results evidence applicability beyond academic benchmarks and highlight strong potential for engineering practice.

6.2 Limitations

Despite the overall robustness, several limitation should be acknowledged.

Finite differences are used only for the derivative of the residual with respect to the excitation frequency, i.e. $\nabla_\omega \boldsymbol{G}$. This quantity is required both to build the `tangent` direction in the predictor and inside the augmented Newton–Raphson corrector. Finite differences add truncation and round-off errors. The effect is stronger near resonances or in stiff regimes. As a result, Newton corrections may need more iterations and the tangent direction can be inaccurate.

The angle-based step size needs careful tuning. If the settings are too conservative, the step keeps shrinking, reaches the minimum allowed size s_{min}, and the computation stops without converging. If the settings are too aggressive, the step becomes too large, the predictor overshoots, and narrow resonances or bifurcations may be missed. Proper calibration keeps steps stable and reduces the number of corrections.

The rotating-frame formulation developed in this work is based on the assumption that gyroscopic effects can be neglected when the rotation axis is orthogonal to the dominant vibration directions. Under this assumption, only centrifugal effects associated with steady rotation are considered, while Coriolis and gyroscopic coupling terms are not included in the formulation.

The rotating-frame formulation developed in this work neglects Coriolis effects. This assumption is suitable for the configurations considered here, where the rotation axis is orthogonal to the dominant vibration directions, resulting in limited velocity-dependent rotational coupling. Accounting for Coriolis effects is therefore identified as an important direction for future developments.

The linear solver MUMPS shows an MPI scaling limit on the largest problems. Beyond a certain number of ranks, adding more MPI does not help and can even slow the run. This behaviour was observed around $1,000,000$ unknowns. For such cases, alternative iterative solver should be evaluated, which was outside the scope of this thesis.

By default, computations are performed in double precision. However, near resonance frequencies the algebraic system becomes severely ill-conditioned and highly sensitive to rounding errors, and in large-scale multiphysics problems accumulated floating-point errors may further degrade the accuracy. In such situations, double precision may fail to ensure convergence, and quadruple precision is required, but this comes at the cost of a substantially higher computational time.

6.3 Future Work

Future work can be structured around two main directions.

On the modelling side, additional physical phenomena could be incorporated, including advanced material models and explicit contact with friction, which are particularly relevant for vibrating multi-part and moving assemblies. The framework could also be extended to coupled multiphysics problems, such as fluid-structure interaction, thermo-mechanical effects, or electro-mechanical behaviour in MEMS devices. In addition, future investigations should address modal interactions involving multiple modes.

On the numerical side, the continuation framework could be enhanced by incorporating bifurcation detection and tracking capabilities, enabling the automatic identification of folds, branch switching, and jump phenomena. Mixed-precision strategies also represent a promising direction, with computations performed in double precision by default and locally promoted to higher precision near strong resonances or ill-conditioned regions. Another perspective consists in exploring adaptive harmonic enrichment strategies, in which higher harmonics are activated only in specific regions of the response where they are required, thereby reducing the overall computational cost. Moreover, large-scale iterative linear solvers should be investigated to overcome the MPI scalability limitations observed with MUMPS. In this context, the iterative solvers available in QUANSCIENT ALLSOLVE should be assessed on very large-scale problems, with particular attention paid to parallel scalability, memory consumption, and solution accuracy.

This page intentionally left blank.

Bibliography

[1] C. Grenat. "Nonlinear Normal Modes and multi-parametric continuation of bifurcations : Application to vibration absorbers and architectured MEMS sensors for mass detection". Theses. Université de Lyon, Oct. 2018. URL: `https://theses.hal.science/tel-02094162`.

[2] S. Zhong et al. "Review of Vibration Analysis and Structural Optimization Research for Rotating Blades". In: *Journal of Marine Science and Application* 24 (May 2024). DOI: `10.1007/s11804-024-00405-6`.

[3] G. Kerschen et al. "Past, Present and Future of Nonlinear System Identification in Structural Dynamics". In: *Mechanical Systems and Signal Processing* 20 (2006), pp. 505–592. DOI: `10.1016/j.ymssp.2005.04.008`.

[4] M. Krack, L. Salles, and F. Thouverez. "Vibration Prediction of Bladed Disks Coupled by Friction Joints". In: *Archives of Computational Methods in Engineering* 24 (2017), pp. 589–636. DOI: `10.1007/s11831-016-9183-2`.

[5] C. Touzé and A. Vizzaccaro. "Nonlinear Normal Modes as Invariant Manifolds for Model Order Reduction". In: *Model Order Reduction for Design, Analysis and Control of Nonlinear Vibratory Systems*. Ed. by C. Touzé and A. Frangi. Cham: Springer Nature Switzerland, 2025, pp. 59–116. DOI: `10.1007/978-3-031-67499-0_2`.

[6] A. Vizzaccaro et al. "Direct computation of nonlinear mapping via normal form for reduced-order models of finite element nonlinear structures". In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113957. ISSN: 0045-7825. DOI: `https://doi.org/10.1016/j.cma.2021.113957`.

[7] O. Thomas, A. Sénéchal, and J.-F. Deü. "Hardening/softening behavior and reduced order modeling of nonlinear vibrations of rotating cantilever beams". In: *Nonlinear Dynamics* 86.2 (2016), pp. 1293–1318. DOI: `10.1007/s11071-016-2965-0`.

[8] S. Quaegebeur et al. "Exploiting internal resonances in nonlinear structures with cyclic symmetry as a mean of passive vibration control". In: *Mechanical Systems and Signal Processing* 178 (2022), p. 109232. ISSN: 0888-3270. DOI: `10.1016/j.ymssp.2022.109232`.

[9] N. Di Palma et al. "Nonlinear Harmonic Analysis of a Blade Model Subjected to Large Geometrical Deflection and Internal Resonance". In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 58691. American Society of Mechanical Engineers. 2019, V07BT35A017. DOI: `10.1115/GT2019-91213`.

[10] J. S. Hesthaven, D. I. Gottlieb, and S. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Vol. 21. Cambridge University Press, 2007. DOI: `10.1017/CBO9780511618352`.

[11] T. Detroux et al. "The harmonic balance method for bifurcation analysis of large-scale nonlinear mechanical systems". In: *Computer Methods in Applied Mechanics and Engineering* 296 (2015), pp. 18–38. ISSN: 0045-7825. DOI: `https://doi.org/10.1016/j.cma.2015.07.017`.

[12] M. Krack and J. Gross. *Harmonic Balance for Nonlinear Vibration Problems*. 1st ed. 2019. Mathematical Engineering. Cham: Springer International Publishing, 2019. ISBN: 3-030-14023-7. DOI: `10.1007/978-3-030-14023-6`.

[13] J. Blahoš. "Parallel Harmonic Balance Method for Analysis of Nonlinear Mechanical Systems". PhD thesis. Imperial College London, Department of Mechanical Engineering, 2022. DOI: `10.25560/100834`.

[14] M. Krack et al. "The Tribomechadynamics Research Challenge: Confronting blind predictions for the linear and nonlinear dynamics of a thin-walled jointed structure with measurement results". In: *Mechanical Systems and Signal Processing* 224 (2025), p. 112016. ISSN: 0888-3270. DOI: `10.1016/j.ymssp.2024.112016`.

[15] A. Opreni et al. "Analysis of the Nonlinear Response of Piezo-Micromirrors with the Harmonic Balance Method". In: *Actuators* 10 (Jan. 2021), p. 21. DOI: `10.3390/act10020021`.

[16] D. Giagopoulos et al. "Finite Element Model Developed and Modal Analysis of Large Scale Steam Turbine Rotor: Quantification of Uncertainties and Model Updating". In: *Proceedings of UNCECOMP 2017*. Jan. 2017, pp. 32–44. DOI: `10.7712/120217.5349.16898`.

[17] J. Blahoš et al. *Parallel Harmonic Balance Method Towards Very Large Scale Systems*. July 2022. DOI: `10.13140/RG.2.2.29200.89600`.

[18] L. Salles. "Etude de l'usure par fretting sous chargements dynamiques dans les interfaces frottantes : application aux pieds d'aubes de turbomachines". Thesis. Ecole Centrale de Lyon ; Université Technique d'Etat Bauman de Moscou, Dec. 2010. URL: `https://theses.hal.science/tel-00600613`.

[19] M.I. Friswell et al. *Dynamics of Rotating Machines*. en. Cambridge Aerospace Series, Vol. 26. Cambridge: Cambridge University Press, 2010. ISBN: 9780521850162.

[20] J. Yuan et al. "On an improved adaptive reduced-order model for the computation of steady-state vibrations in large-scale non-conservative systems with friction joints". In: *Nonlinear Dynamics* 103 (2021), pp. 3283–3300. DOI: `10.1007/s11071-020-05890-2`.

[21] A. Bajaj and C.M. Krousgrill. "Nonlinear System Resonance Phenomena". In: *Encyclopedia of Vibration*. Ed. by S. Braun. Oxford: Elsevier, 2001, pp. 928–943. ISBN: 978-0-12-227085-7. DOI: `10.1006/rwvb.2001.0043`.

[22] Q. Bai and Y. Bai. "Finite Element Analysis of In Situ Behavior". In: *Subsea Pipeline Design, Analysis, and Installation*. Ed. by Q. Bai and Y. Bai. Boston: Gulf Professional Publishing, 2014, pp. 171–185. ISBN: 978-0-12-386888-6. DOI: `10.1016/B978-0-12-386888-6.00008-0`.

[23] G. Kerschen et al. "Nonlinear normal modes, Part I: A useful framework for the structural dynamicist". In: *Mechanical Systems and Signal Processing* 23.1 (2009). Special Issue: Non-linear Structural Dynamics, pp. 170–194. ISSN: 0888-3270. DOI: `https://doi.org/10.1016/j.ymssp.2008.04.002`.

[24] M. Volvert and G. Kerschen. "Phase Resonance Nonlinear Modes of Mechanical Systems". In: *Journal of Sound and Vibration* 511 (Oct. 2021), p. 116355. ISSN: 0022-460X. DOI: `10.1016/j.jsv.2021.116355`.

[25] G. Bourdouch. "Master thesis and internship: Machine learning for experimental bifurcation analysis - Integration internship". Unpublished master's thesis. MA thesis. Liège, Belgique: Université de Liège, 2025. URL: `https://matheo.uliege.be/handle/2268.2/23382`.

[26] R. M. Rosenberg. “The Normal Modes of Nonlinear n-Degree-of-Freedom Systems”. In: *Journal of Applied Mechanics* 29 (1962), pp. 7–14. DOI: `10.1115/1.3636501`.

[27] R. M. Rosenberg. “On Normal Mode Vibrations”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 60.3 (1964), pp. 595–611. DOI: `10.1017/S0305004100038081`.

[28] R. M. Rosenberg. “On Nonlinear Vibrations of Systems with Many Degrees of Freedom”. In: *Advances in Applied Mechanics* 9 (1966), pp. 156–243. DOI: `10.1016/S0065-2156(08)70008-5`.

[29] M. Peeters et al. “Nonlinear Normal Modes, Part II: Toward a Practical Computation Using Numerical Continuation Techniques”. In: *Mechanical Systems and Signal Processing* 23.1 (2009). Special Issue: Non-linear Structural Dynamics, pp. 195–216. DOI: `10.1016/j.ymssp.2008.04.003`.

[30] S. Shaw and C. Pierre. “Non-linear Normal Modes and Invariant Manifolds”. In: *Journal of Sound and Vibration* 150.1 (1991), pp. 170–173. DOI: `10.1016/0022-460X(91)90412-D`.

[31] A. Opreni et al. *High Order Direct Parametrisation of Invariant Manifolds for Model Order Reduction of Finite Element Structures: Application to Generic Forcing Terms and Parametrically Excited Systems*. Preprint on Research Square. 2022. DOI: `10.21203/rs.3.rs-1359763/v1`.

[32] A. Vizzaccaro et al. *High Order Direct Parametrisation of Invariant Manifolds for Model Order Reduction of Finite Element Structures: Application to Large Amplitude Vibrations and Uncovering of a Folding Point*. Preprint on arXiv. 2021. arXiv: `2109.10031 [math.NA]`.

[33] Z. Yan et al. “Harmonic Balance Methods: A Review and Recent Developments”. In: *Computer Modeling in Engineering & Sciences* 137.2 (2023), pp. 1419–1459. ISSN: 1526-1506. DOI: `10.32604/cmes.2023.028198`.

[34] J. Lu, X. Zhao, and S. Yamada. *Harmonic Balance Finite Element Method: Applications in Nonlinear Electromagnetics and Power Systems*. John Wiley & Sons, Sept. 2016, pp. 1–271. ISBN: 9781118975763. DOI: `10.1002/9781118975770`.

[35] Peng Li et al. “Parallel circuit simulation: A historical perspective and recent developments”. In: *Foundations and Trends® in Electronic Design Automation* 5.4 (2012), pp. 211–318. DOI: `10.1561/9781601985156`.

[36] L. Junge et al. “A New Harmonic Balance Approach Using Multidimensional Time”. In: *Journal of Engineering for Gas Turbines and Power* 143.8 (2021), p. 081007. DOI: `10.1115/1.4049698`.

[37] J. Crespo, R. Corral, and J. Pueblas. “An Implicit Harmonic Balance Method in Graphics Processing Units for Oscillating Blades”. In: *Journal of Turbomachinery* 138.3 (2016), p. 031001. DOI: `10.1115/1.4031918`.

[38] H. Poincaré. “Mémoire sur les courbes définies par une équation différentielle”. fr. In: *Journal de Mathématiques Pures et Appliquées* 3e série, 7 (1881), pp. 375–422. URL: `http://www.numdam.org/item/JMPA_1881_3_7__375_0/`.

[39] Klein. “Neue Beiträge zur Riemann’schen Functionentheorie. (Mit 2 lithographirten Tafeln)”. In: *Mathematische Annalen* 21 (1883), pp. 141–218. URL: `http://eudml.org/doc/157059`.

[40] S. Bernstein. “Sur la généralisation du problème de Dirichlet”. In: *Mathematische Annalen* 69 (), pp. 82–136. URL: `https://api.semanticscholar.org/CorpusID:117210948`.

[41] E. L. Allgower and K. Georg. “Numerical continuation methods - an introduction”. In: *Springer Series in Computational Mathematics*. 1990. URL: `https://api.semanticscholar.org/CorpusID:123074585`.

[42] H. B. Keller. “Numerical solution of bifurcation and nonlinear eigenvalue problems”. In: *Applications of Bifurcation Theory*. Ed. by P. Rabinowitz. New York: Academic Press, 1977, pp. 359–384.

[43] M.A. Crisfield. “A fast incremental/iterative solution procedure that handles “snap-through””. In: *Computers Structures* 13.1 (1981), pp. 55–62. ISSN: 0045-7949. DOI: `https://doi.org/10.1016/0045-7949(81)90108-5`.

[44] B. Pan et al. “Double-Homotopy Method for Solving Optimal Control Problems”. In: *Journal of Guidance, Control, and Dynamics* 39.8 (2016). Cited by: 99, pp. 1706–1720. DOI: `10.2514/1.G001553`.

[45] N. Damil and M. Potier-Ferry. “A New Method to Compute Perturbed Bifurcations: Application to the Buckling of Imperfect Elastic Structures”. en. In: *International Journal of Engineering Science* 28.9 (1990), pp. 943–957. ISSN: 0020-7225. DOI: `10.1016/0020-7225(90)90043-I`.

[46] Department of Computer Science and Software Engineering, Concordia University. *AUTO: A Tool for Bifurcation Analysis*. `http://indy.cs.concordia.ca/auto/`. Accessed: 22 April 2025.

[47] Y. A. Kuznetsov et al. *MatCont and CL_MatCont: Continuation Toolboxes in MATLAB*. Manual. Accessed: 22 April 2025. Utrecht University. Aug. 2019. URL: `https://webspace.science.uu.nl/~kouzn101/NBA/ManualMatcontAug2019.pdf`.

[48] R. Arquier, B. Cochelin, and C. Vergez. “Manlab : Logiciel de continuation interactif”. In: *7^e Colloque National en Calcul des Structures*. HAL ID: hal-01813076, version 1. CSMA. Giens, France, May 2005. URL: `https://hal.science/hal-01813076`.

[49] Z. Ahsan et al. “Methods of continuation and their implementation in the COCO software platform with application to delay differential equations”. In: *Nonlinear Dynamics* 107.3 (Mar. 2022). Accessed: 22 April 2025, pp. 3181–3243. DOI: `10.1007/s11071-021-06841-1`.

[50] M. Krack and J. Gross. *NLvib – MATLAB tool for the analysis of nonlinear vibrations*. `https://www.ila.uni-stuttgart.de/nlvib/`. Institut für Luftfahrtantriebe, Universität Stuttgart. Available under GPL-3.0 License. Source code: `https://github.com/maltekrack/NLvib`. 2019.

[51] NOLISYS. *NI2D® Software – Nonlinear Vibration Analysis*. `https://nolisys.com/?page_id=97`. Accessed: 22 April 2025.

[52] A. G. Salinger et al. *LOCA 1.0: Library of Continuation Algorithms – Theory and Implementation Manual*. Tech. rep. SAND2002-0396. Accessed: 22 April 2025. Albuquerque, NM and Livermore, CA, USA: Sandia National Laboratories, Mar. 2002. URL: `https://www.osti.gov/servlets/purl/800778`.

[53] J.-C. Tournier, V. Donde, and Z. Li. “Comparison of Direct and Iterative Sparse Linear Solvers for Power System Applications on Parallel Computing Platforms”. In: *17th Power Systems Computation Conference (PSCC 2011)*. Jan. 2011.

[54] D. Padua. *Encyclopedia of Parallel Computing*. eng. 1st ed. 2011. Springer reference. New York, NY: Springer US, 2011. ISBN: 0-387-09766-X.

[55] MUMPS Technologies and NEC Corporation. *MUMPS – MUltifrontal Massively Parallel sparse direct Solver : version optimisée pour NEC SX-Aurora TSUBASA*. `https://sx-aurora.info/fileadmin/user_upload/Image/Applications/mumps/MUMPS-NEC-AURORA.pdf`. Présentation de la prise en charge de l’architecture vectorielle NEC SX-Aurora TSUBASA mise en œuvre par directive et optimisation du code source, avec délégation transparente de l’appel METIS vers le CPU. 2022.

[56] X. S. Li and J. W. Demmel. "SuperLU_DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems". In: *ACM Transactions on Mathematical Software* 29.2 (2003), pp. 110–140. DOI: 10.1145/779359.779361.

[57] PETSc Developers. *PETSc: Portable, Extensible Toolkit for Scientific Computation*. 2023. URL: https://www.mcs.anl.gov/petsc.

[58] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*. Society for Industrial and Applied Mathematics, 2015. ISBN: 978-1-61197-405-8.

[59] A. Lieu et al. "A non-overlapping Schwarz domain decomposition method with high-order finite elements for flow acoustics". In: *Computer Methods in Applied Mechanics and Engineering* 369 (2020), p. 113223. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.113223.

[60] A. Jackson, M. S. Campobasso, and M. H. Baba-Ahmadi. "On the Parallelization of a Harmonic Balance Compressible Navier-Stokes Solver for Wind Turbine Aerodynamics". In: *Proceedings of the ASME Turbo Expo 2011: Turbine Technical Conference and Exposition*. ASME, 2011, pp. 747–761. DOI: 10.1115/GT2011-45306.

[61] W. Dong and P. Li. "A Parallel Harmonic-Balance Approach to Steady-State and Envelope-Following Simulation of Driven and Autonomous Circuits". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28.4 (2009), pp. 490–501. DOI: 10.1109/TCAD.2009.2014000.

[62] Z. Wang et al. "A Fast Parallel Algorithm Based on GPU for Harmonic Balance Method". In: *2024 IEEE 12th Asia-Pacific Conference on Antennas and Propagation (APCAP)*. 2024, pp. 1–2. DOI: 10.1109/APCAP62011.2024.10881553.

[63] J. Blahoš et al. "Parallel Harmonic Balance Method for Analysis of Nonlinear Dynamical Systems". In: *Proceedings of the ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition. Volume 11: Structures and Dynamics: Structural Mechanics, Vibration, and Damping; Supercritical CO2*. Milan, Italy: ASME, 2020. DOI: 10.1115/GT2020-15392.

[64] A. Saponaro et al. "Parallel Computation of the Nonlinear Forced Response of a Bladed Disk With Friction Contacts Using the FETI Method". In: *Journal of Engineering for Gas Turbines and Power* 148.1 (2026), p. 011021. DOI: 10.1115/1.4069613.

[65] M. Geradin and D. J. Rixen. "Mechanical Vibrations: Theory and Application to Structural Dynamics". English. In: Third. New York: Wiley, 2015. Chap. 4. ISBN: 9781118900208.

[66] C. Touzé, M. Vidrascu, and D. Chapelle. "Direct finite element computation of non-linear modal coupling coefficients for reduced-order shell models". In: *Computational Mechanics* 54 (2014), pp. 567–580. DOI: 10.1007/s00466-014-1006-4.

[67] A. Martin et al. "Reduced-order Modeling of Geometrically Nonlinear Rotating Structures Using the Direct Parametrisation of Invariant Manifolds". In: *Journal of Theoretical, Computational and Applied Mechanics* (2023). DOI: 10.46298/jtcam.10430.

[68] J. F. Hall. "Problems encountered from the use (or misuse) of Rayleigh damping". In: *Earthquake Engineering & Structural Dynamics* 35.5 (2006), pp. 525–545. DOI: 10.1002/eqe.541.

[69] A. K. Chopra. *Dynamics of Structures: Theory and Applications to Earthquake Engineering*. 4th. Prentice Hall, 2012.

[70] E. M. Dawson and Z. Cheng. "Maxwell Damping: An Alternative to Rayleigh Damping". In: *Geo-Extreme 2021*. Proceedings. Reston, VA: American Society of Civil Engineers, Nov. 2021, pp. 34–45. DOI: 10.1061/9780784483701.004.

[71] V. Jaumouillé, J.-J. Sinou, and B. Petitjean. "An adaptive harmonic balance method for predicting the nonlinear dynamic responses of mechanical systems—Application to bolted structures". In: *Journal of Sound and Vibration* 329.19 (2010), pp. 4048–4067. ISSN: 0022-460X. DOI: https://doi.org/10.1016/j.jsv.2010.04.008.

[72] T. Cameron and J. Griffin. "An Alternating Frequency/Time Domain Method for Calculating the Steady-State Response of Nonlinear Dynamic Systems". In: *Journal of Applied Mechanics* 56.1 (1989), pp. 149–154. DOI: 10.1115/1.3176036.

[73] C. Kelley. "Solving nonlinear equations with Newton's method". eng. In: *Scitech book news* 27.4 (2003). ISSN: 0196-6006.

[74] L. Woiwode et al. "Comparison of two algorithms for Harmonic Balance and path continuation". In: *Mechanical Systems and Signal Processing* 136 (2020), p. 106503. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2019.106503.

[75] E. Phipps and A. Salinger. *Bordered Matrix Methods for Large-Scale Stability Analysis using LOCA*. Trilinos User's Group Meeting, November 8, 2006. Nov. 2006. URL: https://www.osti.gov/servlets/purl/1264048.

[76] B. Krauskopf et al. *Numerical Continuation Methods for Dynamical Systems: Path following and boundary value problems*. eng. 1. Aufl. Understanding Complex Systems. Dordrecht: Canopus Publishing Limited, 2007. ISBN: 9781402063558.

[77] S.A.H.A.E. Tabatabaei, A. Malek, and E. Shakour. "Numerical Solution for the Nonlinear Eigenvalue Problems in Bifurcation Points". en. In: *Applied Mathematics and Computation* 195.2 (2008), pp. 397–401. ISSN: 0096-3003. DOI: 10.1016/j.amc.2007.04.117.

[78] J. A. Dahlke and R. A. Bettinger. "Practical implementation of pseudo-arclength continuation to ensure consistent path direction". In: *Acta Astronautica* 215 (2024), pp. 205–216. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2023.12.007.

[79] R. Seydel. *Practical bifurcation and stability analysis*. eng. 3rd ed. Interdisciplinary applied mathematics ; 5. New York: Springer, 2010. ISBN: 9781441917393.

[80] C. B. Haselgrove. "The Solution of Non-Linear Equations and of Differential Equations with Two-Point Boundary Conditions". In: *The Computer Journal* 4.3 (1961), pp. 255–259. DOI: 10.1093/comjnl/4.3.255.

[81] J. B. Evans, P. Xue, and B. Liu. "Analysis and Implementation of Variable Step Size Adaptive Algorithms". In: *IEEE Transactions on Signal Processing* 41.8 (1993), pp. 2517–2535. DOI: 10.1109/78.229885.

[82] J. Yoon and B. Kim. "Step-length control analysis with a multi-staged clutch damper model in a torsional vibration problem using harmonic balance method". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 230.12 (2016), pp. 1968–1977. DOI: 10.1177/0954406215586588.

[83] T. Kubatschek and A. Förster. "Investigation of existing and new approaches to step size control in a continuation framework". In: *Computers and Structures* 313 (2025), p. 107747. DOI: 10.1016/j.compstruc.2025.107747.

[84] A. Fayezioghani, B. Vandoren, and L. J. Sluys. "Performance-based step-length adaptation laws for path-following methods". In: *Computers Structures* 223 (2019), p. 106100. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2019.07.009.

[85] L. Salles et al. "Continuation Techniques for Analysis of Whole Aeroengine Dynamics with Imperfect Bifurcations and Isolated Solutions". In: *Nonlinear Dynamics* 86 (Nov. 2016). DOI: 10.1007/s11071-016-3003-y.

[86] F. J. Muñoz-Almaraz et al. "Continuation of Periodic Orbits in Conservative and Hamiltonian Systems". In: *Physica D: Nonlinear Phenomena* 181.1 (2003), pp. 1–38. DOI: 10.1016/S0167-2789(03)00097-6.

[87] R. Arquier et al. "Two Methods for the Computation of Nonlinear Modes of Vibrating Systems at Large Amplitudes". In: *Computers & Structures* 84.24 (2006), pp. 1565–1576. DOI: 10.1016/j.compstruc.2006.01.011.

[88] Quanscient Oy. *Quanscient Allsolve: Cloud-native multiphysics and quantum-enhanced simulation software*. Accessed March 26, 2025. 2025. URL: https://quanscient.com.

[89] A. Halbach. *Sparselizard – the User-Friendly Finite Element C++ Library*. English. Documentation for Sparselizard – the user-friendly finite element C++ library. Dec. 2017. URL: https://www.sparselizard.org.

[90] OpenBLAS Contributors. *OpenBLAS: An optimized BLAS library*. 2023. URL: https://www.openblas.net.

[91] SLEPc Team. *SLEPc: Scalable Library for Eigenvalue Problem Computations*. 2023. URL: http://slepc.upv.es.

[92] C. Geuzaine and J.-F. Remacle. *Gmsh: A 3D finite element mesh generator with built-in pre- and post-processing facilities*. 2023. URL: https://gmsh.info.

[93] SAF Research Group. *PyHarm: Python Toolbox for Harmonic Balance Analysis*. https://pyharm-saf.readthedocs.io/en/latest/. Accessed: 22 April 2025.

[94] C. Kadapa. "A simple extrapolated predictor for overcoming the starting and tracking issues in the arc-length method for nonlinear structural mechanics". In: *Engineering Structures* 234 (2021), p. 111755. ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2020.111755.

[95] Kenneth G. Dickson, C. T. Kelley, and Ioannis G. Kevrekidis. "Condition Estimates for Pseudo-Arclength Continuation". In: *SIAM Journal on Numerical Analysis* 44.1 (2006), pp. 382–403. DOI: 10.1137/060654384.

[96] EDF R&D (Électricité de France Research & Development). *Code_Aster User Documentation Version 3.04.105*. v3.04.105. Consulté le 17 nov. 2025. Code_Aster Development Team. 2023. URL: https://codeaster.gitlab.io/doc/docaster/manuals/man_v/v3/v3.04.105/index.html.

[97] V. Puzyrev and J. Cela. "A review of block Krylov subspace methods for multisource electromagnetic modelling". In: *Geophysical Journal International* 202 (June 2015), pp. 1241–1252. DOI: 10.1093/gji/ggv216.