POLITECNICO DI TORINO

Corso di Laurea
in Matematica per l'Ingegneria

Tesi di Laurea

An application of the FETI method to a friction damped system

Relatori
prof. Stefano Berrone
prof. Stefano Zucca
firma dei relatori

....
....

Candidato
Lorenzo Cuseri

firma del candidato

....

Anno Accademico 2024-2025

Ai miei genitori

Summary

A domain decomposition algorithm based on the FETI method is developed for the solution of friction contact problems in structural dynamics, coupled with the framework of vibration analysis. After a thorough theoretical discussion, we follow with the description of a practical algorithm with MATLAB implementation and the presentation of results showing the physical phenomena involved and proving the effectiveness of parallel scalability in the 1D case.

Acknowledgements

I would like to thank professors Stefano Berrone and Stefano Zucca for the kind and patient support and everyone that helped me during the arduous time of development of this work.

Contents

List of Tables

List of Figures

List of Algorithms

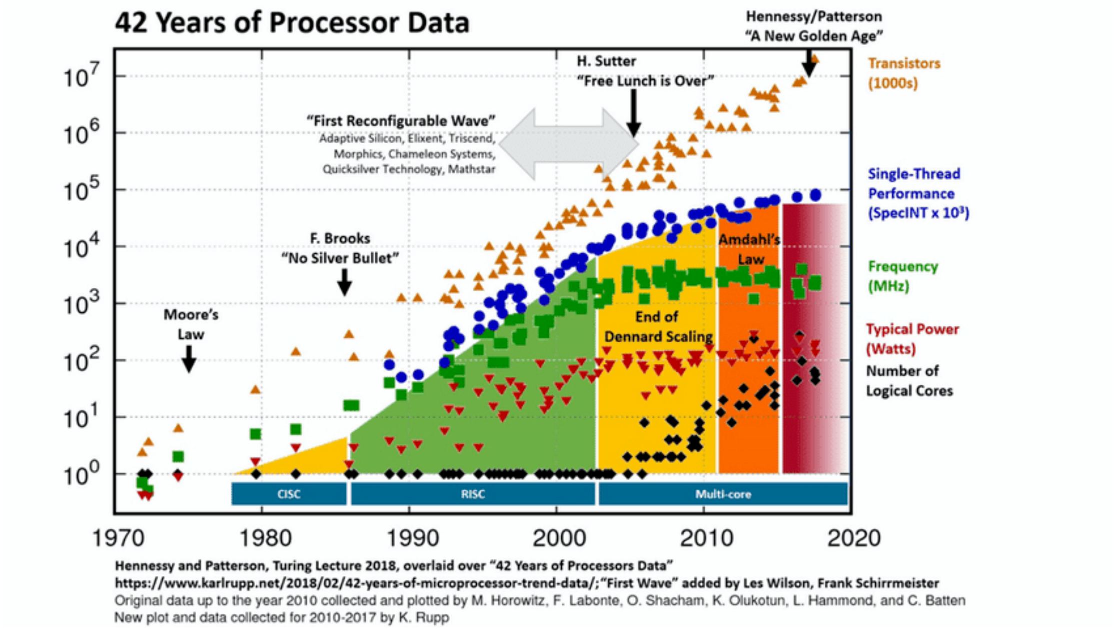
Chapter 1

Introduction

Since the early 2000's, the improvement in computing performance has been kept steady mostly by the increase in the number of cores per processor, rather than higher clock speeds. This implies that any category of software such as visualization algorithms, data storage, operating systems, scientific computing, etc. must be designed with parallelism in mind. *Domain Decomposition Methods* (DDMs) fit into this context as a family of techniques that enables algebraic problems in scientific computing to be solved on multi-core machines; as opposed to parallel implementations of *direct methods* (such as *LU* or Cholesky decompositions), DDMs seek to formulate subproblems that can be naturally distributed on multiple workers for high efficiency and scalability. The two main branches of DDMs are *Schwarz* and *Neumann-Neumann/FETI*; Schwarz algorithms (at least in the original implementation) rely on *overlapping subdomains* whereas for Neumann-Neumann/FETI require subdomains to meet only at *interfaces.* FETI (Finite Element Tearing and Interconnecting), in particular, ensures a weak continuity of the solution at the interfaces thanks to the introduction of *Lagrange multipliers* and will be our point of reference for achieving a working parallel algorithm in MATLAB.

This thesis aims to present a theoretical foundation and a parallel framework that has wide applicability in structural dynamics while following a specific 1D case study which serves to outline in detail various aspects of the implementation. The problem in question is the modelling of the dynamic behaviour of structures with friction damping, a feature commonly used in turbine blade design to reduce vibration amplitudes at resonance and, consequently, to prevent high cycle fatigue failures. These types of structures typically experience periodic stresses which renders inherently appropriate to treat the equations of motion with a *Fourier series expansion* and the *Harmonic Balance Method* (HBM) in place of time-advancing schemes. Speeding up the solution of this equations through a parallel algorithm is especially useful when running parametric analysis and optimizations during the design phase. The main challenge here is represented by combining the nonlinear behaviour of contact forces and the HBM framework with a parallel algorithm (FETI) that originally was not integrated with them.

This work is divided in two main parts: in Chapter 2 the structural dynamics problem is developed from physics first principles to the algebraic equations of motion, a rigorous variational formulation is then presented and followed by the *finite element discretization*



Hennessy and Patterson, Turing Lecture 2018, overlaid over "42 Years of Processors Data"
https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/; "First Wave" added by Les Wilson, Frank Schirrmeister
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Figure 1.1. History of computational performance and its main drivers over time. *Source: Muralidhar, Rajeev et al. (2020). Energy Efficient Computing Systems: Architectures, Abstractions and Modeling to Techniques and Standards.*

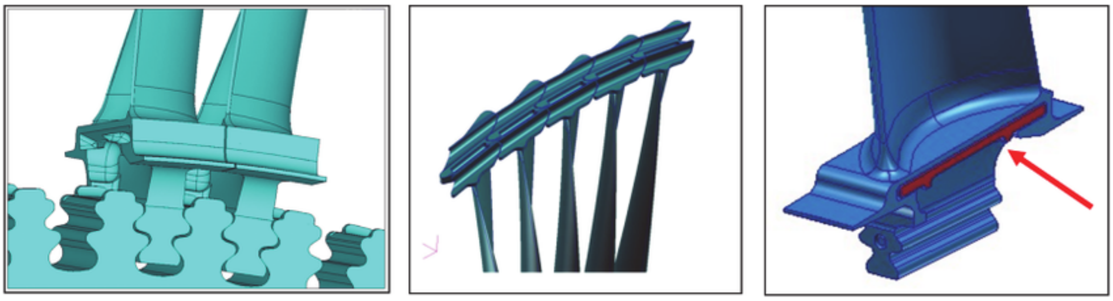


Figure 1.2. Different types of friction contacts in turbine bladed disks: blade root joints, blade shrouds and underplatform dampers. *Source: Firrone, Christian and Zucca, Stefano (2011). Modelling Friction Contacts in Structural Dynamics and its Application to Turbine Bladed Disks.*

(FEM) via the *Galerkin method*; Chapter 3 outlines the FETI method and discusses its application to the previously found results, then proposes an algorithm in MATLAB that assembles and solves the given problem. Some sections to be highlighted as of particular importance are represented by Section 2.5 where the Harmonic Balance Method is introduced and its contributions both to the physical understanding of the system and the algebraic manipulation are described, Sections 3.1 and 3.2 contain the key steps that allow the parallel formulation to be efficiently adapted to the case study and, finally, Section 3.5 presents results that prove the parallel scalability of the algorithm and the behaviour of the main potential design and computational parameters.

With regard to the main sources used to develop this discussion, we underscore as of primary relevance [11] and [19] as the starting point for the solution algorithm in serial form and for the modeling of friction contacts, then [8] and [14] among others for the FETI method and its generalization and developments, and [3], [4] and [16] as fundamental support for the theoretical development of, respectively, the variational formulation, the finite element method and algorithms for solving nonlinear systems of equations.

The main topics presented in this work (friction contact problems, FEM, HBM and FETI methods), if taken one by one, are already covered extensively in the academic literature, but much less so when looking for all of them used together. Contact problems, for instance, have been the frequent subject of the application of dedicated FETI methods in the frictionless case, leading to the FETI-C and the FETI-DPC algorithms (see [1]); this is worth remembering in order to single out the difference with the friction-damped case analyzed here and, consequently, to avoid confusion with FETI-C. The work found in [7], instead, follows a similar path to ours regarding the theoretical development and, although limited to the static behaviour, it reaches further in terms of implementation, showing 3D case studies for Hertzian contact (with linear and non-linear solutions) and Coulomb friction. Two of the same authors of [7] are also involved in an extensive overview on the subject of algorithms for contact problems in the book [6], that the reader may refer to for a wider array of models. Finally, regarding Harmonic Balance Methods, their range of applications and literature is vast, including, but not limited to, nonlinear vibration problems arising from structures, fluids and electric circuits; in order to limit ourselves to a single source that gives a comprehensive theoretical foundation, we think the reader may find useful [13].

Chapter 2

The structural dynamics problem

2.1 The wave equation

In this section we outline the problem from the very basic equilibrium of forces to the mathematical formalization which will be necessary to obtain the equations of motion in matrix form. Some readers may find this process redundant and obvious, but it does nonetheless lay a solid foundation upon which the algorithm will be finally built and it helps identifying the physical meaning of the quantities involved.

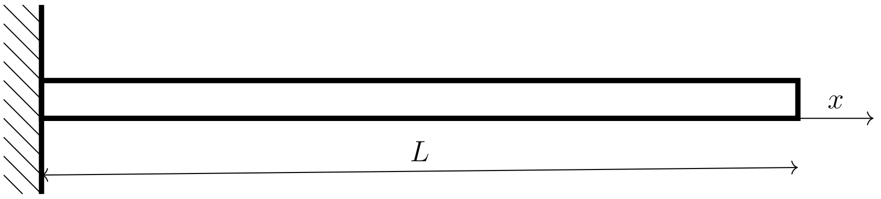


Figure 2.1. 1D clamped beam or cantilever.

The sample problem in question is the 1D beam with one clamped end (commonly known as *cantilever*, Figure 2.1) and the variable of interest is the axial displacement $\mathbf{u}(x,t)$ at every point along the beam when this is subjected to some external axial force per unit lenght $\mathbf{f}(x,t)$. Every infinitesimal element of the beam of lenght dx is subject to the normal (axial) forces exchanged with the neighbouring elements plus the internal damping force and, in general, the external force (Figure 2.2). Furthermore we define the constant parameters $A > 0$ as the cross section of the beam and $\rho > 0$ as the mass density.

Following is the equilibrium of forces along x given by *Newton's second law*:

$$dm\, a_x = \sum F_x$$

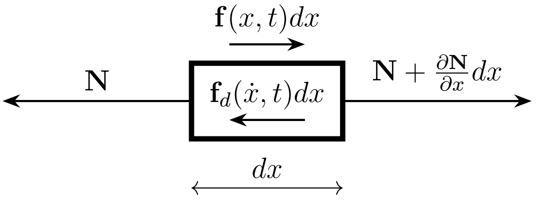

$\mathbf{f}(x,t)dx$
$\mathbf{N}$
$\mathbf{f}_d(\dot{x},t)dx$
$\mathbf{N}+\frac{\partial \mathbf{N}}{\partial x}dx$
dx

Figure 2.2. Forces acting on the beam element.

$$\rho A dx\,\frac{\partial^2 u}{\partial^2 t} = N + \frac{\partial N}{\partial x}dx - N - f_d dx + f dx$$

$$\rho A\frac{\partial^2 u}{\partial^2 t} = \frac{\partial N}{\partial x} - f_d + f \tag{2.1}$$

Now, in order to be entirely expressed in terms of u, Equation (2.1) needs a constitutive relation for N linking forces and strains; such relation is found in *Hooke's law of elasticity* (2.2), where the *Young's modulus* $E > 0$ is introduced [4]:

$$N = EA\frac{\partial u}{\partial x} \tag{2.2}$$

Furthermore, Equation (2.1) needs an expression for the damping term: in general, it is hard to model analitically all the physical phenomena that contribute to it, thus, the most often used technique to introduce damping is called *proportional damping* (see [4]), which relies explicitly on the stiffness and mass matrices for its definition; this definition can be applied only after the discretization in space is performed (see Section 2.4). Before getting to that point and for the discussion that follows, we will rely on a simple viscous-type damper definition stating $f_d = \gamma\frac{\partial u}{\partial t}$, with γ a positive constant. An additional component of damping consists of nonlinear friction damping which acts as an external force that, for now, remains implicit in the term f. With all of this in mind, substituting in (2.1), we arrive at the damped and forced wave equation:

$$\rho A\frac{\partial^2 u}{\partial^2 t} + \gamma\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(EA\frac{\partial u}{\partial x}\right) + f$$

$$\rho A\frac{\partial^2 u}{\partial^2 t} + \gamma\frac{\partial u}{\partial t} - EA\frac{\partial^2 u}{\partial^2 x} = f \tag{2.3}$$

Now, this result, derived from physical principles, will be passed through the more rigorous lens of the variational formulation, the *Dirichlet* and *Neumann* boundary conditions and the initial condition will be introduced, then appropriate *functional spaces* will be assigned to make each resulting term appropriately defined. The reason to elaborate variational (or weak) formulations of problems based on PDEs is twofold: finding a solution in strong form implies asking for high regularity (such as $u \in C^2(\overline{\Omega})$), this is often impossible to achieve in general terms and also cannot account for real-world boundary conditions that are non-differentiable (a simple example of which could be a concentrated load); furthermore, the main discretization techniques for reaching numerical approximations of the solution are built around weak formulations (see [3] and [17]).

$$\begin{cases} \rho A \dfrac{\partial^2 u}{\partial^2 t} + \gamma \dfrac{\partial u}{\partial t} - EA \dfrac{\partial^2 u}{\partial^2 x} = f \quad \text{in } \Omega = (0, L),\ \forall\, t \in \mathcal{T} = (0, T] \\ u = 0 \quad \text{for } \partial\Omega_D = \Gamma_D = 0,\ \forall\, t \in \mathcal{T} \\ \dfrac{\partial u}{\partial x} = 0 \quad \text{for } \partial\Omega_N = \Gamma_N = L,\ \forall\, t \in \mathcal{T} \\ u(0) = \mu_0 \quad \text{in } \Omega \\ \dfrac{\partial u}{\partial t}(0) = \nu_0 \quad \text{in } \Omega \end{cases} \tag{2.4}$$

We proceed towards the formulation of problem (2.4) in a weak sense first by multiplying both sides of the equation by a test function $v \in V = H^1_{0,D}(\Omega)$[*] and integrating in space over Ω:

$$\int_\Omega \rho A \frac{\partial^2 u}{\partial^2 t} v \, d\Omega + \int_\Omega \gamma \frac{\partial u}{\partial t} v \, d\Omega - \int_\Omega EA \frac{\partial^2 u}{\partial^2 x} v \, d\Omega = \int_\Omega f v \, d\Omega \qquad \forall\, t \in \mathcal{T}$$

Then, an *integration-by-parts* is performed on the third term on the left-hand side where the boundary conditions imply that the term evaluated at the boundaries vanishes:

$$\int_\Omega \rho A \frac{\partial^2 u}{\partial^2 t} v \, d\Omega + \int_\Omega \gamma \frac{\partial u}{\partial t} v \, d\Omega - \cancelto{0}{EA \frac{\partial u}{\partial x} v \Big|_{\Gamma_D = 0}^{\Gamma_N = L}} + \int_\Omega EA \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, d\Omega = \int_\Omega f v \, d\Omega \qquad \forall\, t \in \mathcal{T}$$

$$\int_\Omega \rho A \frac{\partial^2 u}{\partial^2 t} v \, d\Omega + \int_\Omega \gamma \frac{\partial u}{\partial t} v \, d\Omega + \int_\Omega EA \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, d\Omega = \int_\Omega f v \, d\Omega \qquad \forall\, t \in \mathcal{T} \tag{2.5}$$

In order for (2.5) to make sense (i.e. in order for the integrals to exist and be bounded) and in order to achieve the actual weak formulation, the following statements apply:

i. The displacement u must belong to the same space as the test function so that the third term on the left-hand side in (2.5) is of a *bilinear form* in V (actually, the inner product in V):

$$u \in V \Longrightarrow \int_\Omega \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, d\Omega = k \left(u, v\right)_{V,V} \qquad \forall\, t \in \mathcal{T} \tag{2.6}$$

[*] The *Sobolev space* $H^1(\Omega)$ is the subspace of $L^2(\Omega)$ (the space of *square-integrable functions*), with $\Omega \subset \mathbb{R}^n$ open and bounded, of the functions whose first-order partial derivatives, in the distributional sense, belong to $L^2(\Omega)$, i.e.:

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) \ : \ \frac{\partial v}{\partial x_i} \in L^2(\Omega) \ \textit{for}\ 1 \leq i \leq n \right\}$$

The notation "$0, D$" restricts the space to the functions that vanish on the boundary of Ω with homogeneous Dirichlet conditions.

ii. Looking at (2.3), we see how the second derivative applied to u (an *elliptic operator*) implies that $\frac{\partial^2 u}{\partial^2 x}$ and, therefore, it is natural that also $\frac{\partial^2 u}{\partial^2 t}$, $\frac{\partial u}{\partial t}$ and f all belong to the *dual space* of V, $V' = H^{-1}(\Omega)$:

$$\frac{\partial^2 u}{\partial^2 t}, \frac{\partial u}{\partial t}, f \in V' \quad \forall t \in \mathcal{T} \tag{2.7}$$

iii. From the previous statement, it follows that the first and second term on the left-hand side and the term of the right-hand side of (2.5) can be written as *duality products*:

$$\langle \frac{\partial^2 u}{\partial^2 t}, v\rangle_{V',V},\ \langle \frac{\partial u}{\partial t}, v\rangle_{V',V},\ \langle f, v\rangle_{V',V} \quad \forall t \in \mathcal{T} \tag{2.8}$$

Until now, we always specified the expressions to hold $\forall t \in \mathcal{T}$ without making assumptions on the regularity of u, $\frac{\partial^2 u}{\partial^2 x}$, etc. as they vary in time; we resolve this point now by integrating (2.5) in time and discussing the validity of the result:

$$\int_0^T\int_\Omega \rho A \frac{\partial^2 u}{\partial^2 t} v \, d\Omega \, dt + \int_0^T\int_\Omega \gamma \frac{\partial u}{\partial t} v \, d\Omega \, dt + \int_0^T\int_\Omega EA \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \, d\Omega \, dt = \int_0^T\int_\Omega f v \, d\Omega \, dt \tag{2.9}$$

We ask first for the boundedness of the third term on the left-hand side and, using the definition of the norm in $V = H^1_{0,D}(\Omega)$ which is $\|v\|_V = \left(\int_\Omega \|\nabla v\|_W^2 \, d\Omega\right)^{1/2}$[†]:

$$\left| \int_0^T\int_\Omega \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \, d\Omega \, dt \right| < \infty \quad \Longrightarrow \quad \int_0^T \|u(t)\|_V^2 \, dt < \infty$$

Which is equivalent to asking that the norm of the function u which takes $t \to u(t) \in V$ is square-integrable:

$$\|u\|_V^2 \in L^2(0,T) \quad \Longrightarrow \quad u \in L^2(0,T;V) \tag{2.10}$$

In (2.10) we introduced what is called a *Bochner space* [3], a concept which can be extended in a similar manner to all the terms analyzed previously:

$$\frac{\partial^2 u}{\partial^2 x}, \frac{\partial^2 u}{\partial^2 t}, \frac{\partial u}{\partial t}, f \in L^2(0,T;V')$$

$$v \in L^2(0,T;V)$$

Finally, we ask for some regularity for the initial conditions [3]:

$$\mu_0 \in V = H^1_{0,D}(\Omega)$$

$$\nu_0 \in W = L^2(\Omega)$$

Combining all the results seen so far, we can fully express (2.4) as a variational problem:

[†] Where we are using $W = L^2(\Omega)$.

Variational Problem 2.1.1 *Find* $u \in L^2(0,T;V)$ *with* $\frac{\partial u}{\partial t} \in L^2(0,T;V')$ *and* $\frac{\partial^2 u}{\partial^2 t} \in L^2(0,T;V')$ *such that*

$$\begin{cases} \rho A \, \langle \dfrac{\partial^2 u}{\partial^2 t}, v\rangle_{V',V} + \gamma \, \langle \dfrac{\partial u}{\partial t}, v\rangle_{V',V} + EA \, k\,(u,v)_{V,V} = \langle f, v\rangle_{V',V} \quad \forall\, v \in L^2(0,T;V) \\ u(0) = \mu_0 \in V \\ \dfrac{\partial u}{\partial t}(0) = \nu_0 \in W \end{cases}$$

It can be proven [3] that the requirements summarized in Problem 2.1.1 lead to *existence and uniqueness* of the solution u.

2.2 Discretization in space of the wave equation

The analytical treatment of PDEs is often limited to proving the existence and uniqueness of the solution, without explicitly finding it; the wave equation (2.3) would be an exception in this sense, being often solved through Fourier analysis or with a separation of variables in the form $u(x,t) = \phi(x)\eta(t)$ while asking for f and the boundary conditions to be somewhat "simple". Unfortunately, we need to treat the general case where the forcing term is allowed to be nonlinear, therefore requiring an approximate method such as the finite-element method. This section will outline the discretization in space using the *Galerkin method* and it will describe the assembly of the mass and stiffness matrices: at the core, the goal is to find a solution which can be represented in a finite-dimensional vector space allowing computations to be performed efficiently and effectively on a processor.

We introduce V_δ subspace of V with finite dimension $N_\delta \geq 2$ and characterized by a discretization parameter $\delta > 0$ representing, for example, the lenght of each element (equally spaced grid) or a representative element size (generic grid). In our sample problem, V_δ is defined on $\Omega = (0,L)$ and the corresponding interval $I = [0,L]$ is subdivided in $N = N_\delta - 1$ sub-intervals, thus, for example, identifying $\delta = \frac{L}{N}$ in the equally spaced case; we require the functions $v \in V_\delta$ to be polynomials of degree 1 ($\mathbb{P}_1$) on each sub-interval I_i and to be continuous on I:

$$V_\delta(\Omega) = \{v \in C^0([0,L]) \;\; : \;\; v|_{I_i} \in \mathbb{P}_1(I_i),\, 1 \leq i \leq N\} \tag{2.11}$$

Thus leading to the discrete variational (Galerkin) problem:

Variational Problem 2.2.1 *Find* $u_\delta \in L^2(0,T;V_\delta)$ *with* $\frac{\partial u_\delta}{\partial t} \in L^2(0,T;V_\delta')$ *and* $\frac{\partial^2 u_\delta}{\partial^2 t} \in L^2(0,T;V_\delta')$ *such that*

$$\begin{cases} \rho A \, \langle \dfrac{\partial^2 u_\delta}{\partial^2 t}, v_\delta\rangle_{V',V} + \gamma \, \langle \dfrac{\partial u_\delta}{\partial t}, v_\delta\rangle_{V',V} + EA \, k\,(u_\delta,v_\delta)_{V,V} = \langle f, v_\delta\rangle_{V',V} \quad \forall\, v_\delta \in L^2(0,T;V_\delta) \\ u_\delta(0) = \mu_{0,\delta} \in V_\delta \\ \dfrac{\partial u_\delta}{\partial t}(0) = \nu_{0,\delta} \in W_\delta \end{cases}$$

It can be proven ([17]) that the requirements summarized in the Galerkin Problem 2.2.1 lead to *existence and uniqueness* of the solution u_δ and that $u_\delta \to u$ for $\delta \to 0$. Furthermore, the finite dimensionality allows every $v_\delta \in V_\delta$ to be expressed as a linear combination of N_δ functions, basis of the vector space, which means that there is a unique representation of v_δ as a (column) vector in $\mathbb{R}^{N_\delta}$:

$$V_\delta = \operatorname{vect}\{\varphi_1, \ldots, \varphi_{N_\delta}\} \quad \Longrightarrow \quad v_\delta = \sum_{i=1}^{N_\delta} v_i \varphi_i \quad \Longleftrightarrow \quad \mathbf{v} = (v_i)_{1\le i\le N_\delta} \in \mathbb{R}^{N_\delta}$$

By substituting this result in Problem 2.2.1 we reach the representation in matrix form (duality products are interpreted as inner products)[‡]:

$$\rho A(\frac{\partial^2}{\partial^2 t}\sum_{j=1}^{N_\delta} u_j\varphi_j, \varphi_i) + \gamma(\frac{\partial}{\partial t}\sum_{j=1}^{N_\delta} u_j\varphi_j, \varphi_i) + EA\,k(\sum_{j=1}^{N_\delta} u_j\varphi_j, \varphi_i) = (f, \varphi_i) \;\; \forall\, 1\le i\le N_\delta \tag{2.12}$$

$$\begin{cases} \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \quad \forall\, t \in \mathscr{T} \\ \mathbf{u}(0) = \boldsymbol{\mu}_0 \\ \dot{\mathbf{u}}(0) = \boldsymbol{\nu}_0 \end{cases} \tag{2.13}$$

Where we defined $\mathbf{M} = m_{ij} = \rho A\,(\varphi_j, \varphi_i)$ (the *mass matrix*), $\mathbf{C}$ (the *damping matrix*), $\mathbf{K} = k_{ij} = EA\,k(\varphi_j, \varphi_i)$ (the *stiffness matrix*) and $\mathbf{f} = \langle f, \varphi_i\rangle_{V',V}$ while omitting the partial derivative operators since, at this point, the variation in space no longer appears in the discretized system explicitly.

Despite many choices are possible for the basis $(\varphi_1, \ldots, \varphi_{N_\delta})$, perhaps the most convenient is the following piecewise $\mathbb{P}_1$ *lagrangian basis* [17] (where we define $h_i = \int_{I_i} dx = x_{i+1} - x_i$):

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h_{i-1}} & \text{if } x \in I_{i-1} \\ \dfrac{x_{i+1} - x}{h_i} & \text{if } x \in I_i \\ 0 & \text{otherwise} \end{cases} \qquad 1 < i < N_\delta \tag{2.14}$$

$$\varphi_1(x) = \begin{cases} \dfrac{x_2 - x}{h_1} & \text{if } x \in I_1 \\ 0 & \text{otherwise} \end{cases}, \quad \varphi_{N_\delta}(x) = \begin{cases} \dfrac{x - x_{N_\delta - 1}}{h_{N_\delta - 1}} & \text{if } x \in I_{N_\delta - 1} \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

As can be seen in Figure 2.3, each $\varphi_i(x)$ is a piecewise linear function that takes value 1 at node i and 0 at all the others implying that the coefficients v_i coincide with the values of v_δ at each node effectively making this an interpolation basis. The space V_δ endowed with the basis $(\varphi_1, \ldots, \varphi_{N_\delta})$ over the grid $G = \{I_i\}_{1\le i\le N}$ with nodes in $x_1 < \cdots < x_i < \cdots < x_{N_\delta}$ is what defines every *finite element*.

[‡] From this point and until the end of the chapter, the mathematical language does shift from a variational lens to an algbraic one.

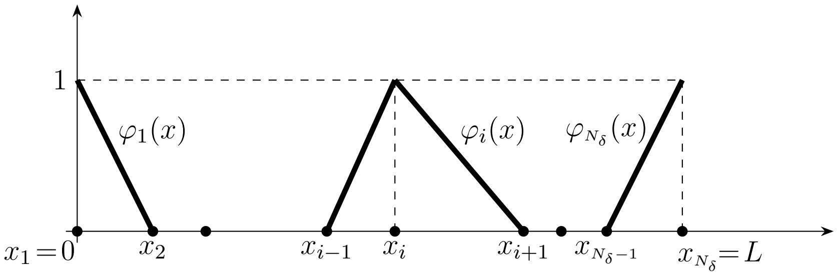


Figure 2.3. The generic basis function φ_i and the basis functions φ_1 and φ_{N_δ} at the edge of the domain.

Some care must be taken when treating boundary conditions: every node that lies on the Dirichlet border Γ_D does not represent a *degree of freedom* since the value of u is already given, thus the basis function φ_1 is eliminated since it insists on the node $x_1 \in \Gamma_D$. The Neumann border Γ_N, instead, does not reduce the degrees of freedom of the system, but it only adds a contribution to the equations in weak form which, in our case, is nil, as seen in Equation (2.5). This means that the actual number of degrees of freedom (N_{dof}) is one less than the number of grid nodes N_δ and equals the number of elements N in which the domain is subdivided; it follows that the system (2.13) has matrices of $\mathbb{R}^{N_{dof}\times N_{dof}}$ and vectors of $\mathbb{R}^{N_{dof}}$ ($N_{dof} = N_\delta - 1 = N$). The i-th node on the grid will map onto the $(i-1)$-th matrix row/column.

The matrices $\mathbf{M}$ and $\mathbf{K}$ are built by assembling the contributions of each possible pair (φ_j, φ_i) to the respective m_{ij} and k_{ij} entries. Looking at Figures 2.4 and 2.5 it is easy to understand that, for each row i, only columns $j = i-1, i, i+1$ give a non-zero contribution implying that: *a)* the matrices are both sparse, tri-diagonal and symmetric and *b)* we can analitically write down the 3 possible outcomes of the computation in the non-zero entries.

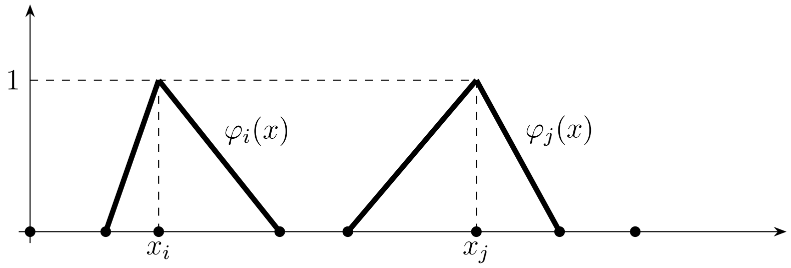


Figure 2.4. A case where the pair of basis functions *implies no contribution* to the matrices.

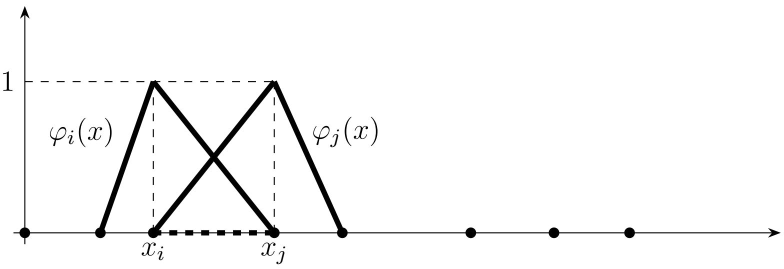


Figure 2.5. A case where the pair of basis functions *does imply a contribution* to the matrices over the element between nodes x_i and x_j.

2.3 Assembly of the discretized system

Starting from the stiffness matrix $\mathbf{K}$, we can trace back the definition of the underlying bilinear form in (2.6) and therefore get the following:

$$k_{ij} = EA \int_{\Omega} \frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\, d\Omega = EA \int_I \varphi_j' \varphi_i' \, dx$$

As previously stated, there are only 3 cases where the integral is non-zero ($j = i{-}1, i, i{+}1$):

$$k_{ij} = EA \begin{cases} \int_{I_{i-1}} \left(\frac{x_i - x}{h_{i-1}}\right)' \left(\frac{x - x_{i-1}}{h_{i-1}}\right)' dx & j = i-1 \\ \int_{I_{i-1}} \left(\frac{x - x_{i-1}}{h_{i-1}}\right)' \left(\frac{x - x_{i-1}}{h_{i-1}}\right)' dx + \int_{I_i} \left(\frac{x_{i+1} - x}{h_i}\right)' \left(\frac{x_{i+1} - x}{h_i}\right)' dx & j = i \\ \int_{I_i} \left(\frac{x - x_i}{h_i}\right)' \left(\frac{x_{i+1} - x}{h_i}\right)' dx & j = i+1 \end{cases}$$

$$k_{ij} = EA \begin{cases} \int_{I_{i-1}} -\frac{1}{h_{i-1}} \frac{1}{h_{i-1}} dx = -\frac{1}{h_{i-1}^2} \int_{I_{i-1}} dx & j = i-1 \\ \int_{I_{i-1}} \frac{1}{h_{i-1}} \frac{1}{h_{i-1}} dx + \int_{I_i} \left(-\frac{1}{h_i}\right)\left(-\frac{1}{h_i}\right) dx = \frac{1}{h_{i-1}^2} \int_{I_{i-1}} dx + \frac{1}{h_i^2} \int_{I_i} dx & j = i \\ \int_{I_i} \frac{1}{h_i} \left(-\frac{1}{h_i}\right) dx = -\frac{1}{h_i^2} \int_{I_i} dx & j = i+1 \end{cases}$$

$$k_{ij} = \begin{cases} -\frac{EA}{h_{i-1}} & j = i-1 \\ \frac{EA}{h_{i-1}} + \frac{EA}{h_i} & j = i \\ -\frac{EA}{h_i} & j = i+1 \end{cases} \tag{2.16}$$

If the grid is equally spaced, i.e. $h_i = h \; \forall \, i$, Equation (2.16) simplifies into:

$$k_{ij} = \frac{EA}{h} \begin{cases} -1 & j = i-1 \\ \;\;2 & j = i \\ -1 & j = i+1 \end{cases} \tag{2.17}$$

This holds except for $i = N_{dof}$, which would map onto the interval I_{N_δ} that does not exist, therefore leaving only the contribution of $I_{i-1} = I_{N_\delta - 1}$, as can be seen in matrix form:

$$\mathbf{K} = \frac{EA}{h} \begin{bmatrix} 2 & 1 & 0 & \cdots & 0 \\ 1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & 1 & 2 & 1 \\ 0 & \cdots & 0 & 1 & 1 \end{bmatrix} \tag{2.18}$$

For the mass matrix, we can start by tracing back its definition when we stepped from (2.12) into the equation in matrix form (2.13):

$$m_{ij} = \rho A \int_\Omega uv \, d\Omega = \rho A \int_I \varphi_j \varphi_i \, dx$$

Which leads to similar computations as seen for the stiffness matrix, only that the derivatives of the basis functions do not appear:

$$m_{ij} = \rho A \begin{cases} \displaystyle\int_{I_{i-1}} \left(\frac{x_i - x}{h_{i-1}} \right) \left(\frac{x - x_{i-1}}{h_{i-1}} \right) dx & j = i-1 \\ \displaystyle\int_{I_{i-1}} \left(\frac{x - x_{i-1}}{h_{i-1}} \right) \left(\frac{x - x_{i-1}}{h_{i-1}} \right) dx + \int_{I_i} \left(\frac{x_{i+1} - x}{h_i} \right) \left(\frac{x_{i+1} - x}{h_i} \right) dx & j = i \\ \displaystyle\int_{I_i} \left(\frac{x - x_i}{h_i} \right) \left(\frac{x_{i+1} - x}{h_i} \right) dx & j = i+1 \end{cases}$$

$$m_{ij} = \rho A \begin{cases} \displaystyle -\frac{1}{h_{i-1}^2} \int_{I_{i-1}} (x_i - x)(x - x_{i-1}) \, dx & j = i-1 \\ \displaystyle \frac{1}{h_{i-1}^2} \int_{I_{i-1}} (x - x_{i-1})^2 \, dx + \frac{1}{h_i^2} \int_{I_i} (x_{i+1} - x)^2 \, dx & j = i \\ \displaystyle -\frac{1}{h_i^2} \int_{I_i} (x - x_i)(x_{i+1} - x) \, dx & j = i+1 \end{cases}$$

$$m_{ij} = \begin{cases} \dfrac{\rho A}{6} h_{i-1} & j = i-1 \\ \rho A \dfrac{h_{i-1} + h_i}{3} & j = i \\ \dfrac{\rho A}{6} h_i & j = i+1 \end{cases} \tag{2.19}$$

If the grid is equally spaced, i.e. $h_i = h \; \forall \, i$, Equation (2.19) simplifies into:

$$m_{ij} = \frac{\rho A h}{6} \begin{cases} 1 & j = i - 1 \\ 4 & j = i \\ 1 & j = i + 1 \end{cases} \tag{2.20}$$

This holds except for $i = N_{dof}$, which would map onto the interval I_{N_δ} that does not exist, therefore leaving only the contribution of $I_{i-1} = I_{N_\delta - 1}$, as can be seen in matrix form:

$$\mathbf{M} = \frac{\rho A h}{6} \begin{bmatrix} 4 & 1 & 0 & \cdots & 0 \\ 1 & 4 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & 1 & 4 & 1 \\ 0 & \cdots & 0 & 1 & 2 \end{bmatrix} \tag{2.21}$$

Finally for the vector of known terms $\mathbf{f}$, we can state the following (notice how, if f was defined as a force per-unit-lenght as in Equation (2.1), now f_i are forces):

$$f_i = \int_\Omega f v \, d\Omega = \int_I f \varphi_i \, dx = \int_{I_{i-1}} f \varphi_i \, dx + \int_{I_i} f \varphi_i \, dx$$

The result of this integral depends on the definition of f; for example, if we can approximate f as an element-wise constant function:

$$f_i = \overline{f}_{i-1} \int_{I_{i-1}} \varphi_i \, dx + \overline{f}_i \int_{I_i} \varphi_i \, dx = \frac{1}{2} \left(\overline{f}_{i-1} h_{i-1} + \overline{f}_i h_i \right) \tag{2.22}$$

Alternatively, we might want to define the entries f_i directly as localized forces acting at *specific nodes of the 1D grid*, as will be done in the following.

2.4 Definition of damping terms

As we hinted at in Section 2.1, the damping matrix $\mathbf{C}$ instead of being assembled as was done for $\mathbf{K}$ and $\mathbf{M}$, is defined through *proportional damping* (also known as *Rayleigh damping*, [4]), as a linear combination of the latter two:

$$\mathbf{C} = \alpha \mathbf{K} + \beta \mathbf{M} \tag{2.23}$$

Looking back at Equation (2.13), it is important to note how the introduction of proportional damping does not change the simple viscous nature of the second term on the left-hand side: for the majority of structures loaded in the elastic range, this is enough to account for most of the underlying complex physical phenomena. This definition not only simplifies the modeling of damping in materials by using just two empirical constants

α and β, but also leads potentially to uncoupled equations when performing the *modal analysis* (see [11]).

The second source of damping in our system is represented by the contribution to $\mathbf{f}$ given by friction contacts at some locations along the domain. Therefore, $\mathbf{f}$ has to be decomposed as the sum of linear external force(s) and the non-linear friction-induced component:

$$\mathbf{f} = \mathbf{f}_l(t) + \mathbf{f}_{nl}(\dot{\mathbf{u}}, \mathbf{u}, t) \tag{2.24}$$

The contact is modeled as in Figure 2.7, where the beam element exchanges tangential and normal forces $\mathbf{T}$ and $\mathbf{N}$ (at a certain grid node) with the surface through respective contact stiffnesses k_t and k_n. The nonlinear behaviour in $\mathbf{T}$ arises when the *Coulomb limit* is met and the system transitions from sticking mode to slipping mode and vice-versa, therefore from a spring relation to a dynamic friction relation:

$$f_{nl} = T = \begin{cases} k_t(u - w) & \text{sticking mode} \\ \text{sgn}(\dot{w})\mu N & \text{slipping mode} \\ 0 & \text{lift-off mode} \end{cases} \tag{2.25}$$

In our case, vertical displacement is not modeled, therefore the modulus of $\mathbf{N}$ is a constant parameter N_0 and lift-off never occours. Depending on the magnitude of N_0 the system frequency response will be closer or further from the linear case and an optimal value for which the amplitude is minimum might be found (see [11]).
The contact model used for this case study and its MATLAB code implementation is derived from [19] and summarized in Figure 2.6: it evaluates the stick/slip transition and assigns the correct magnitude to the force in the time domain involving an implementation of the *Fast Fourier Transform* algorithm and its inverse since, as will be explained in the following Section 2.5, our nonlinear system is treated with Fourier series expansions.

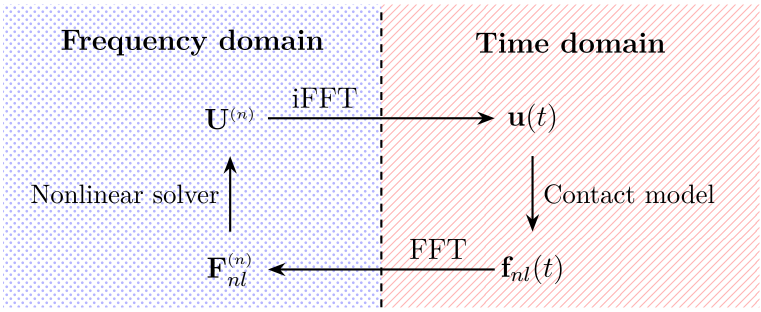


Figure 2.6. Workflow diagram of the contact model.

For further the details on the transition criteria and the implementation of this and other contact models the reader can refer to [11] and [19].

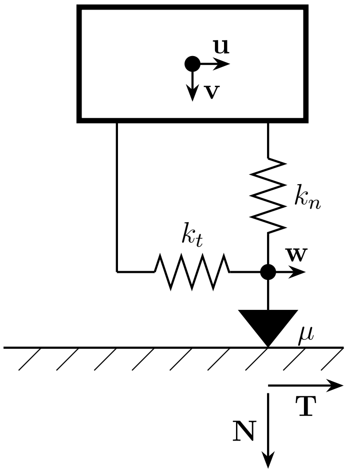


Figure 2.7. Force diagram of the contact model.

2.5 The Harmonic Balance Method

The practical interest of models, such as the one being discussed, is often to analyze systems subject to periodic (harmonic) forces where the explicit transient in time is not nearly as important as the frequency response and the control of resonance phenomena. It is natural, therefore, to treat the system (2.13) using Fourier series expansions truncated at n_H harmonics rather than a discretization in time to be solved via a time-advancing method such as *implicit Euler* or *Crank-Nicholson* (for an in-depth view on the topic of Fourier analysis see [15]). We can now identify a frequency range of interest $\mathscr{F}$ (that is adequate to capture the resonance phenomena of the system) and expand the time-dependent terms as follows:

$$\mathbf{u}(t) \approx \mathbf{U}^{(0)} + \sum_{n=1}^{n_H} \mathbf{U}^{(n)} e^{in\omega t} \quad \forall\, \omega \in \mathscr{F},\, \forall\, t \in \mathscr{T} \tag{2.26}$$

$$\mathbf{f}_l(t) \approx \mathbf{F}_l^{(0)} + \sum_{n=1}^{n_H} \mathbf{F}_l^{(n)} e^{in\omega t} = \mathbf{F}_l^{(1)} e^{i\omega t} \quad \forall\, \omega \in \mathscr{F},\, \forall\, t \in \mathscr{T} \tag{2.27}$$

$$\mathbf{f}_{nl}(\dot{\mathbf{u}}, \mathbf{u}, t) \approx \mathbf{F}_{nl}^{(0)}(\dot{\mathbf{u}}, \mathbf{u}) + \sum_{n=1}^{n_H} \mathbf{F}_{nl}^{(n)}(\dot{\mathbf{u}}, \mathbf{u}) e^{in\omega t} \quad \forall\, \omega \in \mathscr{F},\, \forall\, t \in \mathscr{T} \tag{2.28}$$

Where we assume the external excitation force to be harmonic and, in the following, that the static components $\mathbf{U}^{(0)}$, $\mathbf{F}^{(0)}$ and $\mathbf{F}_{nl}^{(0)}$ can be ignored since the interest is the dynamic response analysis. Substituting in (2.13) we get:

$$\mathbf{M} \sum_{n=1}^{n_H} \mathbf{U}^{(n)} (-n^2\omega^2) e^{in\omega t} + \mathbf{C} \sum_{n=1}^{n_H} \mathbf{U}^{(n)} (in\omega) e^{in\omega t} + \mathbf{K} \sum_{n=1}^{n_H} \mathbf{U}^{(n)} e^{in\omega t} =$$

$$= \mathbf{F}_l e^{i\omega t} + \sum_{n=1}^{n_H} \mathbf{F}_{nl}^{(n)} e^{in\omega t} \quad \forall\, \omega \in \mathscr{F},\, \forall\, t \in \mathscr{T}$$

$$\sum_{n=1}^{n_H}\Big(-n^2\omega^2\mathbf{M}+in\omega\mathbf{C}+\mathbf{K}\Big)\,\mathbf{U}^{(n)}e^{in\omega t}=\mathbf{F}_l e^{i\omega t}+\sum_{n=1}^{n_H}\mathbf{F}_{nl}^{(n)}e^{in\omega t}\quad\forall\,\omega\in\mathscr{F},\,\forall\,t\in\mathscr{T}$$

We can now balance harmonics of the same order on both sides of the equations and, by doing so, eliminate time-dependence and write n_H systems of equations in the frequency domain:

$$\begin{bmatrix}\mathbf{K}_{dyn}^{(1)}(\omega) \\ \mathbf{K}_{dyn}^{(2)}(\omega) \\ \vdots \\ \mathbf{K}_{dyn}^{(n_H)}(\omega)\end{bmatrix}\begin{Bmatrix}\mathbf{U}^{(1)} \\ \mathbf{U}^{(2)} \\ \vdots \\ \mathbf{U}^{(n_H)}\end{Bmatrix}=\begin{Bmatrix}\mathbf{F}_l^{(1)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0}\end{Bmatrix}+\begin{Bmatrix}\mathbf{F}_{nl}^{(1)} \\ \mathbf{F}_{nl}^{(2)} \\ \vdots \\ \mathbf{F}_{nl}^{(n_H)}\end{Bmatrix}\quad\forall\,\omega\in\mathscr{F}\tag{2.29}$$

Were a *dynamic stiffness matrix* was introduced for each harmonic:

$$\mathbf{K}_{dyn}^{(n)}(\omega)=\Big(-n^2\omega^2\mathbf{M}+in\omega\mathbf{C}+\mathbf{K}\Big)\quad\forall\,\omega\in\mathscr{F},\;n=1,\ldots,n_H\tag{2.30}$$

A few observations can be made regarding (2.29):

i. We arrived at a series of non-linear systems of equations in the Fourier coefficients of size $N_{dof}\times N_{dof}$ where the solutions are the vectors of complex amplitudes $\mathbf{U}^{(n)}$.

ii. The systems of equations for each harmonic are *coupled* to one another since every $\mathbf{F}_{nl}^{(n)}$ is a (non-linear) function of $\mathbf{U}^{(1)},\ldots,\mathbf{U}^{(n_H)}$ through $\dot{\mathbf{u}}$ and $\mathbf{u}$ as defined by the contact model.

iii. The number of harmonics to be used n_H depends on the complexity of the system and the accuracy required, keeping in mind that the first handful of harmonics carry most of the energy of any given system; for a 1D beam like the one described, truncating the Fourier series at the first (*fundamental*) harmonic often already gives a good approximation.

iv. In the following sections, for sake of simplicity, the superscripts $^{(n)}$ will be dropped and the rest of the discussion presented is to be considered valid for each and any number of harmonics. Superscripts $^{[p]}$ will instead indicate p-subdomain-specific objects.

Let us introduce now some results of *modal analysis* that are useful both for the physical understanding of the system and for the calculations going forward (as derived from [20] and [4]). Looking at Equations (2.29) and (2.30), if we take only the first harmonic and extract the associated *homogeneous and undamped* system we get:

$$\Big(\mathbf{K}-\omega^2\mathbf{M}\Big)\,\mathbf{U}^{(1)}=\mathbf{0}\tag{2.31}$$

Which has exactly the structure of a *generalized eigenvalue problem* of size $N_{dof}\times N_{dof}$:

$$\left(\mathbf{A}-\lambda\mathbf{B}\right)\mathbf{x}=\mathbf{0}$$

For which we can identify N_{dof} *eigenvalues* $\lambda_j = \omega_j^2$ representing the squared *natural frequencies* of the system and the corresponding *eigenvectors* or *natural modes* $\boldsymbol{\psi}_j$. It is a common practice to normalize the natural modes with respect to the mass matrix:

$$\boldsymbol{\phi}_j = \frac{\boldsymbol{\psi}_j}{\sqrt{\boldsymbol{\psi}_j^T \mathbf{M} \boldsymbol{\psi}_j}}$$

All the m-normalized natural modes (as column vectors) can be assembled into the *m-normalized modal matrix* $\boldsymbol{\Phi}$ (an *orthogonal matrix*) and used to transform the system (2.31) into a diagonal system expressed in *modal coordinates*:

$$\boldsymbol{\Phi} = \begin{bmatrix} & | & \\ \dots & \boldsymbol{\phi}_j & \dots \\ & | & \end{bmatrix} \tag{2.32}$$

Since the damping matrix $\mathbf{C}$ is defined as per Equation (2.23) as a linear combination of $\mathbf{K}$ and $\mathbf{M}$[§], this implies that we can use $\boldsymbol{\Phi}$ to transform the entire dynamic stiffness matrix into a diagonal matrix (for the first harmonic component):

$$\boldsymbol{\Phi}^T \left(-\omega^2 \mathbf{M} + i\omega \mathbf{C} + \mathbf{K}\right) \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{U}^{(1)} = \boldsymbol{\Phi}^T \left(\mathbf{F}_l^{(1)} + \mathbf{F}_{nl}^{(1)}\right) \quad \forall\, \omega \in \mathscr{F}$$

$$\left(-\omega^2 \mathbf{I} + i\omega\, \mathrm{diag}(\mathbf{c}_j) + \mathrm{diag}(\boldsymbol{\omega}_j^2)\right) \mathbf{Q}^{(1)} = \boldsymbol{\Phi}^T \left(\mathbf{F}_l^{(1)} + \mathbf{F}_{nl}^{(1)}\right) \quad \forall\, \omega \in \mathscr{F}$$

$$\mathbf{K}_{mod}^{(1)}(\omega)\mathbf{Q}^{(1)} = \boldsymbol{\Phi}^T \left(\mathbf{F}_l^{(1)} + \mathbf{F}_{nl}^{(1)}\right) \quad \forall\, \omega \in \mathscr{F} \tag{2.33}$$

Where the modal dynamic stiffness matrix $\mathbf{K}_{mod}^{(n)}(\omega)$:

$$\mathbf{K}_{mod}^{(n)}(\omega) = -n^2\omega^2 \mathbf{I} + in\omega\, \mathrm{diag}(\mathbf{c}_j) + \mathrm{diag}(\boldsymbol{\omega}_j^2)$$

and the Fourier coefficients transformed in modal amplitudes $\mathbf{Q}^{(n)} = \boldsymbol{\Phi}^T \mathbf{U}^{(n)}$ have been introduced. If the non-linearity did not exist, one could simply proceed solving system (2.33) in a closed form, inverting the diagonal matrix and analyzing explicitly the influence of every mode on the dynamic response. In our case, despite the absence of a closed form solution, the results (2.32) and (2.33) is still very important since a single eigenvalue/eigenvector computation is reusable to invert efficiently $\mathbf{K}_{mod}^{(n)}(\omega)$ for each harmonic and for each frequency. The solution, translated back in physical coordinates and for any harmonic component using $\mathbf{U}^{(n)} = \boldsymbol{\Phi}\mathbf{Q}^{(n)}$, can be written as:

$$\mathbf{U}^{(n)} = \boldsymbol{\Phi} \left(\mathbf{K}_{mod}^{(n)}(\omega)\right)^{-1} \boldsymbol{\Phi}^T \left(\mathbf{F}_l^{(n)} + \mathbf{F}_{nl}^{(n)}\right) \quad \forall\, \omega \in \mathscr{F},\ n = 1, \dots, n_H \tag{2.34}$$

[§] In the context of modal analysis, proportional damping simplifies into:

$$\mathrm{diag}(\mathbf{c}_j) = \alpha_m \mathrm{diag}(\boldsymbol{\omega}_j^2)$$

With α_m being a *modal damping* factor.

Chapter 3

The parallel solution algorithm

3.1 The FETI method and its application

The original FETI method, introduced around 1990 (the milestone paper being [10]), is a finite element DDM based on non-overlapping subdomains where the compatibility of the displacement field on the shared boundaries is enforced via *Lagrange multipliers.* The following discussion will review some of the results seen in Section 2.1 through the lens of the FETI method showing how naturally the decomposed system of equations will emerge. This domain decomposition method will be adapted to the framework of modal analysis and it will *not* require a preconditioned conjugate projected gradient (PCPG) algorithm for the solution of the interface problem, which is one of the core components of the original FETI method, therefore making it more appropriate to categorize our approach as a *FETI-like* method or, more generically, as a *Lagrange-multiplier-based* method.
We start by splitting via an interface boundary Γ_I our domain Ω in N_s subdomains $\{\Omega_1, \dots, \Omega_{N_s}\}$ on which the local displacements $\{u^{[1]}, \dots, u^{[N_s]}\}$ are defined. In order for this manipulation to still make sense, we use the *isomorphism* between the space $H^1(\Omega)$ and its "decomposed" counterpart $\mathscr{H}^1(\Omega_1, \dots, \Omega_{N_s})$ (sometimes referred to as "broken" space) when the continuity on the boundary Γ_I is imposed (see [5] and [14]):

$$\begin{gathered}\mathscr{H}^1(\Omega_1, \dots, \Omega_{N_s}) = \\ \Big\{(u^{[1]}, \dots, u^{[N_s]}) \in H^1(\Omega_1) \times \dots \times H^1(\Omega_{N_s}) \ : \ u^{[p]} = u^{[q]} \text{ on } \Gamma_I \ \forall p, q = 1, \dots, N_s\Big\}\end{gathered} \tag{3.1}$$

with the boundary Γ_I being defined as:

$$\begin{aligned}\Gamma_{pq} &= \Omega_p \cap \Omega_q \\ \Gamma_I &= \bigcup_{p,q=1}^{N_s} \{\Omega_p \cap \Omega_q\}\end{aligned} \tag{3.2}$$

Expanding on the derivation given in [8], we use the weak form found in Problem 2.1.1 and write the corresponding *energy functional*, setting $u = v \in L^2(0, T; \mathscr{V})$:

$$\mathscr{E}(v) = \rho A \,\langle \frac{\partial^2 v}{\partial^2 t}, v\rangle_{\mathscr{V}', \mathscr{V}} + \gamma \,\langle \frac{\partial v}{\partial t}, v\rangle_{\mathscr{V}', \mathscr{V}} + EA\, k\,(v, v)_{\mathscr{V}, \mathscr{V}} - \langle f, v\rangle_{\mathscr{V}', \mathscr{V}} \tag{3.3}$$

In order to find the solution in the decomposed space, $H^1_{0,D}(\Omega)$ and $L^2(\Omega)$ are replaced by $\mathscr{V} = \mathscr{H}^1_{0,D}(\Omega_1, \dots, \Omega_{N_s})$ and $\mathscr{W} = L^2(\Omega_1, \dots, \Omega_{N_s})$ with definitions equivalent to the one seen in (3.1). The Variational Problem 2.1.1 is transformed into the search of a stationary point of $\mathscr{E}(v)$.

Variational Problem 3.1.1 *Find* $u \in L^2(0,T;\mathscr{V})$ *with* $\frac{\partial u}{\partial t} \in L^2(0,T;\mathscr{V}')$ *and* $\frac{\partial^2 u}{\partial^2 t} \in L^2(0,T;\mathscr{V}')$ *such that*

$$\begin{cases} \mathscr{E}(u) = 0 \\ u(0) = \mu_0 \in \mathscr{V} \\ \dfrac{\partial u}{\partial t}(0) = \nu_0 \in \mathscr{W} \end{cases}$$

This problem is equivalent to solving N_s sub-problems with "local" functionals $\mathscr{E}_p(v^{[p]})$:

$$\begin{aligned} \mathscr{E}_p(v^{[p]}) = \rho A \, \langle \frac{\partial^2 v^{[p]}}{\partial^2 t}, v^{[p]} \rangle_{V_p', V_p} + \gamma \, \langle \frac{\partial v^{[p]}}{\partial t}, v^{[p]} \rangle_{V_p', V_p} \\ + EA \, k \, (v^{[p]}, v^{[p]})_{V_p, V_p} - \langle f^{[p]}, v^{[p]} \rangle_{V_p', V_p} \end{aligned} \tag{3.4}$$

Where now we use $V_p = H^1_{0,D}(\Omega_p)$ and $W_p = L^2(\Omega_p)$. Each subdomain is delimited by a portion Γ_p of Γ and by a portion Γ_{pq} of Γ_I as given in (3.2) ("internal" subdomains where $\Gamma_p = \emptyset$ will be referred to as *floating subdomains*). The resulting set of "local" displacements $\{u^{[1]}, \dots, u^{[N_s]}\}$, constrained by the continuity condition, is equivalent to u.

Variational Problem 3.1.2 *Find* $u \in L^2(0,T;V_p)$ *with* $\frac{\partial u}{\partial t} \in L^2(0,T;V_p')$ *and* $\frac{\partial^2 u}{\partial^2 t} \in L^2(0,T;V_p')$ *such that*

$$\begin{cases} \mathscr{E}_p(u^{[p]}) = 0 \quad \forall \, p = 1, \dots, N_s \\ u^{[p]} - u^{[q]} = 0 \quad \text{on } \Gamma_I \ \forall \, p, q = 1, \dots, N_s, \forall \, t \in \mathscr{T} \\ u^{[p]}(0) = \mu_0^{[p]} \in V_p \\ \dfrac{\partial u^{[p]}}{\partial t}(0) = \nu_0^{[p]} \in W_p \end{cases}$$

A strategy of enforcing multiple equality contraints, which is commonplace in the theory of constrained optimization (see [16]), involves the construction of a *Lagrangian* for the Problem 3.1.2 as the one that follows:

$$\mathscr{L}(u^{[1]}, \dots, u^{[N_s]}, \lambda_{1\,2}, \dots, \lambda_{N_s-1\,N_s}) = \sum_{p=1}^{N_s} \mathscr{E}_p(u^{[p]}) + \sum_{p,q=1}^{N_s} \mathscr{C}(\lambda_{pq}) \tag{3.5}$$

where $\mathscr{C}(\lambda_{pq})$ is used to enforce the corresponding constraint $u^{[p]} - u^{[q]} = 0$ expressed in weak (variational) form. In order find its appropriate definition, we start by multiplying the constraint by a test function μ_{pq} (the *Lagrange multiplier*) and integrating over the interface boundary:

$$\frac{1}{2} \int_{\Gamma_I} \operatorname{tr}(u^{[p]} - u^{[q]}) \, \mu_{pq} \, d\Gamma = 0 \quad \forall \, \mu_{pq} \in S_{pq}', \forall \, t \in \mathscr{T} \tag{3.6}$$

This new term requires the introduction a corresponding functional space S_{pq} in addition to what has been seen in Section 2.1, since it is the first integral performed on the boundary that will provide a nonzero contribution. Now, if we denote by $H^{\frac{1}{2}}(\partial\Omega_p)$ the space of the trace of functions $v \in H^1(\Omega_p)$ on $\partial\Omega_p$ and by $H_{00}^{\frac{1}{2}}(\Gamma_{pq})$ its restriction to only functions that vanish on the portion of "outer" boundary $\partial\Omega_p \backslash \Gamma_{pq} = \Gamma_p$, then the Lagrange multipliers belong to $S'_{pq} = H^{-\frac{1}{2}}(\Gamma_{pq})$, dual space of $S_{pq} = H_{00}^{\frac{1}{2}}(\Gamma_{pq})$. As has been done in Section 2.1, we then ask for a minimal level of regularity in the time domain by introducing the corresponding Bochner space:

$$\frac{1}{2}\langle \mu_{pq}, \mathrm{tr}\,(u^{[p]} - u^{[q]})\rangle_{S'_{pq}, S_{pq}} = 0 \quad \forall\, \mu_{pq} \in L^2(0,T;S'_{pq}) \tag{3.7}$$

This implies that $\mathscr{C}(\lambda_{pq})$ can be defined, for the generic test functions $v^{[p]}$ and $v^{[q]}$, as:

$$\mathscr{C}(\lambda_{pq}) = \frac{1}{2}\langle \lambda_{pq}, \mathrm{tr}\,(v^{[p]} - v^{[q]})\rangle_{S'_{pq}, S_{pq}} \quad \forall\, v^{[p]}, v^{[q]} \in L^2(0,T;V_p) \tag{3.8}$$

Substituting in the Lagrangian we get:

$$\mathscr{L}(u^{[1]}, \dots, u^{[N_s]}, \lambda_{1\,2}, \dots, \lambda_{N_s-1\,N_s}) = \sum_{p=1}^{N_s} \mathscr{E}_p(u^{[p]}) + \frac{1}{2}\sum_{p,q=1}^{N_s} \langle \lambda_{pq}, \mathrm{tr}\,(v^{[p]} - v^{[q]})\rangle_{S'_{pq}, S_{pq}} \tag{3.9}$$

Solving Problem 3.1.2 (in space) is equivalent to finding the *saddle point* of $\mathscr{L}$ and, thus, finding the set of all displacement fields $u^{[p]} \in L^2(0,T;V_p)$ and the set of all Lagrange multipliers $\lambda_{pq} \in L^2(0,T;S'_{pq})$ which satisfy:

$$\begin{aligned} &\mathscr{L}(u^{[1]}, \dots, u^{[N_s]}, \mu_{1\,2}, \dots, \mu_{N_s-1\,N_s}) \\ \le\, &\mathscr{L}(u^{[1]}, \dots, u^{[N_s]}, \lambda_{1\,2}, \dots, \lambda_{N_s-1\,N_s}) \\ \le\, &\mathscr{L}(v^{[1]}, \dots, v^{[N_s]}, \lambda_{1\,2}, \dots, \lambda_{N_s-1\,N_s}) \quad \forall\, v^{[p]} \in L^2(0,T;V_p), \forall\, \mu_{pq} \in L^2(0,T;S'_{pq}) \end{aligned} \tag{3.10}$$

Using the natural requirement for symmetry $\lambda_{pq} = -\lambda_{qp}$, it is easy to reach the following simplified form of Equation (3.9):

$$\mathscr{L}(u^{[1]}, \dots, u^{[N_s]}, \lambda_1, \dots, \lambda_{N_s}) = \sum_{p=1}^{N_s} \mathscr{E}_p(u^{[p]}) + \sum_{p=1}^{N_s} \int_{\Gamma_I} \lambda_p \mathrm{tr}\,(v^{[p]})\; d\Gamma \tag{3.11}$$

Where, for the convenience of term reordering, the following definition has been applied:

$$\lambda_p = \sum_{q=1}^{N_s} \lambda_{pq} \tag{3.12}$$

The Problem 3.1.2 together with the results in (3.7), (3.11) and (3.12) are instrumental for the derivation of the following

Variational Problem 3.1.3 *Find $u \in L^2(0,T;V_p)$ with $\frac{\partial u}{\partial t} \in L^2(0,T;V_p')$ and $\frac{\partial^2 u}{\partial^2 t} \in L^2(0,T;V_p')$ and find $\lambda_{pq} \in L^2(0,T;S_{pq}')$ such that*

$$\begin{cases} \mathscr{E}_p(u^{[p]}) + \displaystyle\int_{\Gamma_I} \lambda_p \mathrm{tr}\,(v^{[p]})\; d\Gamma = 0 \quad \forall\, p = 1,\dots,N_s \\ \langle \mu_{pq}, \mathrm{tr}\,(u^{[p]} - u^{[q]})\rangle_{S_{pq}',S_{pq}} = 0 \quad \text{on } \Gamma_I \;\forall\, p,q = 1,\dots,N_s, \forall\, \mu_{pq} \in L^2(0,T;S_{pq}') \\ u^{[p]}(0) = \mu_0^{[p]} \in V_p \\ \dfrac{\partial u^{[p]}}{\partial t}(0) = \nu_0^{[p]} \in W_p \end{cases}$$

Problem 3.1.3 is actually a *hybrid variation problem*, since it is based on an equilibrium principle (saddle point, see Equations (3.9) and (3.10)) rather than a more typical minimization principle, and the existence and uniqueness of the solution can be proven in the framework of mixed and hybrid problems, for which [2] provides a thorough analysis.

We can now suppose to re-write the corresponding of the Galerkin Problem 2.2.1 in each subdomain, where we now know, thanks to the hybrid variational principle 3.1.3, that an additional boundary term will appear (physically can be thought of as generalized forces exchanged between the neighouring subdomains). Over the discretized geometry, the dimension of $(S_{pq}')_\delta$ is obviously smaller that $(V_p)_\delta$, therefore a mapping is needed. A corresponding additional term is therefore what distinguishes the "local" algebraic system from the original system (2.13):

$$\begin{cases} \mathbf{M}^{[p]}\ddot{\mathbf{u}}^{[p]} + \mathbf{C}^{[p]}\dot{\mathbf{u}}^{[p]} + \mathbf{K}^{[p]}\mathbf{u}^{[p]} = \mathbf{f}^{[p]} - \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\lambda} \quad \forall\, t \in \mathscr{T} \\ \mathbf{u}^{[p]}(0) = \boldsymbol{\mu}_0^{[p]} \\ \dot{\mathbf{u}}^{[p]}(0) = \boldsymbol{\nu}_0^{[p]} \end{cases} \qquad \forall\, p = 1,\dots,N_s \tag{3.13}$$

The column vector $\boldsymbol{\lambda}$ is essentially the collection of all the Lagrange multipliers needed to connect the subdomains along their interface Γ_I and $\mathbf{B}^{[p]}$ is the required mapping operator in the form of a *signed connectivity boolean matrix* that extracts the interface components relevant to the subdomain p, while assigning them the correct sign that maintains overall balance of forces, in accordance to *Newton's third law.* It also can be used to represent algebraically the continuity constraint between subdomains:

$$\sum_{p=1}^{N_s} \mathbf{B}^{[p]}\mathbf{u}^{[p]} = 0 \quad \forall\, p = 1,\dots,N_s \tag{3.14}$$

If, for example, we imagine a 1D domain that has been divided in 4 subdomains all having 5 DOFs, there are 3 interface nodes and, therefore, $\boldsymbol{\lambda}$ will have size 3×1 and every $\mathbf{B}^{[p]}$ will have size 3×5. In this example, the $\mathbf{B}^{[p]}$s will have the following structure:

$$\left(\mathbf{B}^{[1]}\right)^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \text{-1} & 0 & 0 \end{bmatrix}, \left(\mathbf{B}^{[2]}\right)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \text{-1} & 0 \end{bmatrix}, \left(\mathbf{B}^{[3]}\right)^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \text{-1} \end{bmatrix}, \left(\mathbf{B}^{[4]}\right)^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We now assume that we can apply the HBM method seen for the complete domain in Section 2.5 to Equation (3.13), treating $\boldsymbol{\lambda}$ similarly to the nonlinear component of force (2.28):

$$\boldsymbol{\lambda}(t) \approx \boldsymbol{\Lambda}^{(0)} + \sum_{n=1}^{n_H} \boldsymbol{\Lambda}^{(n)} e^{in\omega t} \quad \forall\, \omega \in \mathscr{F},\, \forall\, t \in \mathscr{T}$$

Thus obtaining for the generic harmonic for each subdomain:

$$\begin{gathered}\mathbf{K}_{mod}^{[p]}\mathbf{Q}^{[p]} = \left(\boldsymbol{\Phi}^{[p]}\right)^T \left(\mathbf{F}_l^{[p]} + \mathbf{F}_{nl}^{[p]} - \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\Lambda}\right) \\ \forall\, \omega \in \mathscr{F},\; n = 1, \ldots, n_H,\; p = 1, \ldots, N_s\end{gathered} \tag{3.15}$$

$$\begin{gathered}\sum_{p=1}^{N_s} \mathbf{B}^{[p]}\boldsymbol{\Phi}^{[p]}\mathbf{Q}^{[p]} = 0 \\ \forall\, \omega \in \mathscr{F},\; n = 1, \ldots, n_H\end{gathered} \tag{3.16}$$

The FETI method is then completed by the search for $\boldsymbol{\Lambda}$ as described in the next section.

3.2 Interface problem

In the previous section we understood how shift our view from the algebraic equation for the full system in modal coordinates to a series of algebraic equations of smaller dimension defined in every subdomain. By doing this, a new term was introduced, $\boldsymbol{\Lambda}$, which we are about to define, but first it is necessary to understand the nature of the subdomains involved. Let's imagine that we want to solve Equation (3.15) by inverting the stiffness matrix without any knowledge of its structure, therefore remaining in a general case where the matrix could be singular if the subdomain is floating (i.e. if *rigid body modes* do exist in Ω_p):

$$\mathbf{U}^{[p]} = \boldsymbol{\Phi}^{[p]} \left(\mathbf{K}_{mod}^{[p]}\right)^+ \left(\boldsymbol{\Phi}^{[p]}\right)^T \left(\mathbf{F}_l^{[p]} + \mathbf{F}_{nl}^{[p]} - \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\Lambda}\right) + \boldsymbol{\Phi}^{[p]}\mathbf{R}_{\mathbf{K}}^{[p]}\mathbf{a}^{[p]} \quad \forall\, \omega,\, n,\, p \tag{3.17}$$

Where "+" indicates a *generalized inverse*, $\mathbf{R}_{\mathbf{K}}^{[p]}$ stores the rigid body modes (in other words, it's the *kernel* of $\mathbf{K}_{mod}^{[p]}$) and $\mathbf{a}$ specifies a linear combination of these. There are at most one of them in one dimension, three of them in two dimensions and six of them in three dimensions. Rigid body modes are such that:

$$\left(\mathbf{R}_{\mathbf{K}}^{[p]}\right)^T \mathbf{K}_{mod}^{[p]}\mathbf{Q}^{[p]} = \left(\mathbf{R}_{\mathbf{K}}^{[p]}\right)^T \left(\boldsymbol{\Phi}^{[p]}\right)^T \left(\mathbf{F}_l^{[p]} + \mathbf{F}_{nl}^{[p]} - \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\Lambda}\right) = 0 \tag{3.18}$$

Combining Equation (3.17) with the constraint in Equation (3.16) we get:

$$\begin{aligned}&\sum_{p=1}^{N_s} \mathbf{B}^{[p]}\boldsymbol{\Phi}^{[p]} \left(\mathbf{K}_{mod}^{[p]}\right)^+ \left(\boldsymbol{\Phi}^{[p]}\right)^T \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\Lambda} - \sum_{p=1}^{N_s} \mathbf{B}^{[p]}\boldsymbol{\Phi}^{[p]}\mathbf{R}_{\mathbf{K}}^{[p]}\mathbf{a}^{[p]} \\ &= \sum_{p=1}^{N_s} \mathbf{B}^{[p]}\boldsymbol{\Phi}^{[p]} \left(\mathbf{K}_{mod}^{[p]}\right)^+ \left(\boldsymbol{\Phi}^{[p]}\right)^T \left(\mathbf{F}_l^{[p]} + \mathbf{F}_{nl}^{[p]}\right) \quad \forall\, \omega,\, n\end{aligned} \tag{3.19}$$

This leads, after some algebraic manipulations, to a system called *interface problem* with the following structure:

$$\begin{bmatrix} \mathbf{F}_{\mathbf{I}}^{\star} & \mathbf{G}_{\mathbf{I}} \\ \mathbf{G}_{\mathbf{I}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{\Lambda} \\ \mathbf{a} \end{Bmatrix} = \begin{Bmatrix} \mathbf{d} \\ -\mathbf{e} \end{Bmatrix} \quad \forall\, \omega,\, n \tag{3.20}$$

This general case, though, simplifies significantly given the adoption of the framework of modal analysis, since we have a stiffness matrix that is not only *diagonal*, but also *full-rank* by definition:

$$\mathbf{K}_{mod}^{[p]} = \left(-n^2\omega^2\mathbf{I} + in\omega\,\mathrm{diag}(\mathbf{c}_j) + \mathrm{diag}(\boldsymbol{\omega}_j^2)\right) \quad \forall\, \omega,\, n,\, p$$

This leads to three major advantages:

i. The inverse always exists and is computationally efficient.

ii. The several instances of the matrix corresponding to each frequency in $\mathscr{F}$ can be stored efficiently as columns of a single matrix.

iii. The need to specify rigid body modes and distinguish between floating and non-floating subdomains disappears, since $\mathbf{a} = \emptyset$, simplifying the dynamic equations into:

$$\mathbf{U}^{[p]} = \mathbf{\Phi}\left(\mathbf{K}_{mod}^{[p]}\right)^{-1}\left(\mathbf{\Phi}^{[p]}\right)^{T}\left(\mathbf{F}_{l}^{[p]} + \mathbf{F}_{nl}^{[p]} - \left(\mathbf{B}^{[p]}\right)^{T}\mathbf{\Lambda}\right) \quad \forall\, \omega,\, n,\, p \tag{3.21}$$

and the interface problem into:

$$\mathbf{F}_{\mathbf{I}}^{\star}\mathbf{\Lambda} = \mathbf{d} \quad \forall\, \omega,\, n \tag{3.22}$$

We, therefore, define explicitly the only two components of the system that matter for this discussion:

$$\mathbf{F}_{\mathbf{I}}^{\star} = \sum_{p=1}^{N_s} \mathbf{B}^{[p]}\mathbf{\Phi}\left(\mathbf{K}_{mod}^{[p]}\right)^{-1}\left(\mathbf{\Phi}^{[p]}\right)^{T}\left(\mathbf{B}^{[p]}\right)^{T} \quad \forall\, \omega,\, n \tag{3.23}$$

$$\mathbf{d} = \sum_{p=1}^{N_s} \mathbf{B}^{[p]}\mathbf{\Phi}\left(\mathbf{K}_{mod}^{[p]}\right)^{-1}\left(\mathbf{\Phi}^{[p]}\right)^{T}\left(\mathbf{F}_{l}^{[p]} + \mathbf{F}_{nl}^{[p]}\right) \quad \forall\, \omega,\, n \tag{3.24}$$

Where $\mathbf{F}_{\mathbf{I}}^{\star}$ is effectively a *dual Schur complement.* In terms of software implementation, the assembly of the interface problem does require some degree of interprocessor communication, but the method described leads overall to its significant reduction if compared to classical substructuring methods (see [8] and [9] more details on computational aspects of the complete FETI method). Once Equation (3.22) is solved, all the processors receive the array $\mathbf{\Lambda}$ and can use it to find the solution of the respective local displacements as per Equation (3.17).

3.3 Nonlinear Newton solver

Equation (3.21) gives the solution to be calculated for each frequency in $\mathscr{F}$, for each harmonic, inside every subdomain Ω_p; it isn't, though, a closed-form solution: the contact model described in Section 2.4 is non-linear, which is reflected by the dependency of $\mathbf{F}_{nl}^{[p]}$ from the displacement (see the remark number ii. in Section 2.5). Furthermore, any change in $\mathbf{F}_{nl}^{[p]}$ is reflected also in $\mathbf{\Lambda}$ through $\mathbf{d}$. This facts suggest the use of an iterative method to solve Equation (3.21) and, following the footsteps of [11] and [20], the choice falls on a Newton-like root-finding approach (see [12] for a broad overview of root-finding methods that might suit different case studies). The classical Newton's method seeks to find the solution of a nonlinear system of equations by solving iteratively a sequence of linear systems; if the initial guess x_0 is "close enough" to the solution $x^\star$, then the method shows a superlinear rate of convergence.
Let us consider a vector of nonlinear functions $\mathbb{R}^m \to \mathbb{R}^m$ in the form:

$$\mathbf{r}(\mathbf{x}) = \begin{Bmatrix} \mathbf{r}_1(\mathbf{x}) \\ \mathbf{r}_2(\mathbf{x}) \\ \vdots \\ \mathbf{r}_m(\mathbf{x}) \end{Bmatrix}$$

We are looking for the root $\mathbf{x}^\star$ (if it is not globally unique, a good initial guess is required) that is a solution to the system of nonlinear equations:

$$\mathbf{r}(\mathbf{x}^\star) = \mathbf{0}$$

In order to evaluate the progress of convergence, a scalar-valued function that indicates whether a new iterate k is better or worse than the previous one is required, it is called *merit function* and is commonly defined as:

$$f_k = f(\mathbf{x}_k) = \frac{1}{2} \left\| \mathbf{r}(\mathbf{x}_k) \right\|_2^2$$

At each iteration, a linear system involving the Jacobian matrix $\mathbf{J}_k$ (over $\mathbf{r}_k$)[°] and $\mathbf{r}_k$ is solved to compute the update step $\mathbf{s}_k$:

$$\mathbf{J}_k \mathbf{s}_k = -\mathbf{r}_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$$

[°] In practical terms, each column of the Jacobian is the finite difference derivative of $\mathbf{r}_k$:

$$[\mathbf{J}_k]_{\cdot i} = \frac{\mathbf{r}(\mathbf{x}_k + \varepsilon \mathbf{e}_i) - \mathbf{r}(\mathbf{x}_k)}{\varepsilon} \qquad i = 1, \dots, N_{dof}$$

With ε some small constant and $\mathbf{e}_i$ the standard basis vector.

Where the scalar $\alpha_k > 0$ is the step length resulting from a *line search* algorithm; the value of α_k seeks to maximize the reduction in f_k along the computed direction $\mathbf{s}_k$: ideally, it should be found as a global minimizer of the scalar function:

$$\eta(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{s}_k), \quad \alpha > 0$$

In general, this operation is too expensive to perform, requiring potentially multiple evaluations of f_k and $\mathbf{J}_k$. A more practical approach is to follow an *inexact line search* and accept a value of α which yields an "adequate" reduction in f_k by imposing iteratively a *sufficient decrease condition*[△]:

$$f(\mathbf{x}_k + \alpha\mathbf{s}_k) \leq f(\mathbf{x}_k) + c\alpha(\nabla f_k)^T \mathbf{p}_k$$

Where c is a small positive constant parameter ($c \approx 10^{-4} \div 10^{-1}$). The complete Newton's method used for this work in detailed in Algorithm 1: the line search follows a backtracking approach and has been optimized to re-use the same α between subsequent iterations supposing similar near-optimal values while progressively recovering it towards 1 to speed-up convergence.

Algorithm 1: Newton's Method with Backtracking Line Search.

```
Initialize $\mathbf{x}_0$, $k$, $\alpha = 1$, $\alpha_{lim} \in (0,1)$, $c \in (0,1)$, $\tau \in (0,1)$, $\theta \in (0,1)$;
while $f_k \geq tol$ do
    Solve $\mathbf{J}_k\mathbf{s}_k = -\mathbf{r}_k$ for $\mathbf{s}_k$;
    Evaluate $f(\mathbf{x}_k)$, $f(\mathbf{x}_k + \alpha\mathbf{s}_k)$;
    while $f(\mathbf{x}_k + \alpha\mathbf{s}_k) \leq f(\mathbf{x}_k) + c\alpha(\nabla f_k)^T\mathbf{s}_k$ do
        Set $\alpha \leftarrow \max(\tau\alpha, \alpha_{lim})$;
        Evaluate $f(\mathbf{x}_k + \alpha\mathbf{s}_k)$;
    end
    Set $\alpha_k \leftarrow \alpha$;
    Set $\alpha \leftarrow \min(\alpha_k + \theta, 1)$;
    Set $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{s}_k$;
    Set $k \leftarrow k + 1$;
end
Return $\mathbf{x}^\star \leftarrow \mathbf{x}_k$;
```

In which way does Algorithm 1 map onto our problem? First of all, it is performed in parallel for every subdomain in order to find the local $\mathbf{U}^{[p]}$. This comes with an observation: the vast majority of the literature on the Newton's method and numerical optimization in

[△] Often used in combination with a *curvature condition* as part of the *Wolfe conditions* (see [16]). It is also useful to notice the following result:

$$(\nabla f_k)^T = -\left(\mathbf{J}_k^T\mathbf{r}_k\right)^T$$

general is referred to *real-valued* systems, but the Fourier analysis framework brings with it complex coefficients; for this reason, real and imaginary parts are split into separate vector components (doubling the size). Furthermore, we need to account for all the required harmonics: these get stacked on top of each other in the same column vector to be solved simultaneously bringing the size to $2 \cdot N_{dof}^{[p]} \cdot n_H \times 1$. This can be expressed as:

$$\begin{aligned} \zeta : \quad & \mathbb{C}^{N_{dof}^{[p]} \times n_H} \mapsto \mathbb{R}^{2 \cdot N_{dof}^{[p]} \cdot n_H} \\ \Rightarrow & \mathbf{x}^{[p]} = \zeta(\mathbf{U}^{[p]}) \end{aligned} \tag{3.25}$$

The nonlinear system of equations is re-written as:

$$\mathbf{r}(\mathbf{x}^{[p]}) = \zeta\left(\mathbf{U}^{[p]} - \boldsymbol{\Phi}\left(\mathbf{K}_{mod}^{[p]}\right)^{-1}\left(\boldsymbol{\Phi}^{[p]}\right)^T\left(\mathbf{F}_l^{[p]} + \mathbf{F}_{nl}^{[p]} - \left(\mathbf{B}^{[p]}\right)^T \boldsymbol{\Lambda}\right)\right) \tag{3.26}$$

The Jacobian, apparently, is extremely expensive to calculate, especially when asking for several harmonics, being $\mathbf{J} \in \mathbb{R}^{2 \cdot N_{dof}^{[p]} \cdot n_H} \times \mathbb{R}^{2 \cdot N_{dof}^{[p]} \cdot n_H}$. Fortunately, it is easy to see how its columns are non-trivial only on DOFs where the contact forces are present; it takes instead the structure of an identity matrix everywhere else:

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & \cdots & J_{1\,(2j-1)} & J_{1\,2j} & \cdots & 0 \\ 0 & 1 & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ddots & \vdots & \vdots & \ddots & 0 \\ \vdots & \vdots & \ddots & J_{(2N_{dof}^{[p]} n_H - 1)\,(2j-1)} & J_{(2N_{dof}^{[p]} n_H - 1)\,2j} & \ddots & 0 \\ 0 & \cdots & \cdots & J_{(2N_{dof}^{[p]} n_H)\,(2j-1)} & J_{(2N_{dof}^{[p]} n_H)\,2j} & \cdots & 1 \end{bmatrix} \tag{3.27}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{j-th DOF has friction damping}}$$

3.4 Complete algorithm

Following is Algorithm 2 that presents in pseudocode the main steps towards the solution, given as complex amplitudes $\mathbf{U}^{[p]}$ at every grid node. A 1:1 correspondence is always assumed between each subdomain and the core/worker that processes its information ($N_s = N_{\text{cores}}$). The algorithm can, of course, be generalized to other case studies and different programming languages, although, here we specifically refer to one of the possible parallel constructs available in MATLAB: first a "pool" of N_{cores} parallel workers is created with `parpool()`, then every statement inside the following `spmd` block (which stands for Single Program Multiple Data) gets executed in parallel. In order for the code to function properly, variables are defined in ways that fall in this categories:

i. "Static" variables of general scope, such as numerical and physical parameters, are defined as usual at the beginning of the code and, as soon as they appear inside the `spmd` block, every worker is automatically sent a copy of each of them.

ii. Variables of "localized utility", such as counters, flags and storage of intermediate results, can be defined directly inside the `spmd` block and will independently take up values according to each worker activity.

iii. Variables that define the solution of the DDM by taking different values accordingly to the information of every subdomain, like those corresponding to the vectors and matrices in Equation 3.26 and those related to the grid topology of every subdomain, must be treated as `Composite()` variables, which is a type of *cell array*, an indexed structure, where the i-th cell is used only by the i-th worker. The arrays contained in each cell can be sized independently and, therefore, despite always being preferable to look for a balanced computational load, subdomains containing unequal number of element are handled with no issue.

It is extremely important for the efficiency of the code that **all the parallel computations are executed inside a single `spmd` block**, otherwise the code does run into significant communication overhead. Sometimes inter-worker communication cannot be avoided and in Algorithm 2 the presence of `[*]` indicates its occurrences, but this, if handled with dedicated functions like `spmdPlus()` (the Composite argument is summed over all workers and the result sent to each one), `spmdReceive()` and/or `spmdSend()`, does not impact efficiency like starting and ending multiple `spmd` blocks to combine serial and parallel sections of code. If an operation is inherently serial, such as verbose output or the assembly of the complete solution, it is possible to ask for a single worker (for example the number 1) to perform it while the others wait with a simple `if spmdIndex==1, *Statements*, end` block. Finally, to ensure coordination between workers and avoid communication mismatches or the code stalling, `spmdBarrier` is a useful statement as the last line inside the `while` loop, blocking further execution until all workers reach `spmdBarrier`.

Algorithm 2: Application of the FETI Method to 1D Friction Damped Systems.

```
Define Physical parameters: $E$, $\tau$, $\alpha_m$, $A$, $L$;
Define Contact parameters: $K_t$, $\mu$, $N_0$, $K_n$;
Define Modal analysis parameters: $\mathscr{F}$, $n_H$;
Define FEM parameters: force and boundary condition locations, number of
  elements;
Define Computational parameters: $N_{\text{cores}}$, Newton's method parameters;
Evaluate Topology of subdomains;
Create Pool of parallel workers (spmd);
Start spmd
    Assemble Local linear systems: $\mathbf{K}_{mod}^{[p]}$, $\mathbf{\Phi}^{[p]}$, $\mathbf{F}_l^{[p]}$;
    for $\omega \in \mathscr{F}$ do
        Initialize $\mathbf{x}$;
        Evaluate $\left(\mathbf{K}_{mod}^{[p]}\right)^{-1}$ in $\omega$, $\forall n = 1, \ldots, n_H$ $[*]$;
        Initialize $\mathbf{x}^{[p]}$, $k$, $\alpha^{[p]}$, $\alpha_{lim}$, $c$, $\tau$, $\theta$;
        while $f_k^{[p]} \geq tol \ \wedge \ k < k_{max}$ do
            Function Find $\mathbf{r}_k^{[p]}$ and $f_k^{[p]}$ :
                Find Contact Forces $\mathbf{F}_{nl}^{[p]}$ through the Contact Model;
                Solve $\mathbf{F}_\mathbf{I}^\star \mathbf{\Lambda} = \mathbf{d}$ for $\mathbf{\Lambda}$;
                Return $\mathbf{r}_k^{[p]}$ and $f_k^{[p]}$;
            end
            Assemble $\mathbf{J}_k^{[p]}$ $[*]$;
            Solve $\mathbf{J}_k \mathbf{s}_k^{[p]} = -\mathbf{r}_k^{[p]}$ for $\mathbf{s}_k^{[p]}$;
            Find $\alpha_k^{[p]}$ through the Backtracking Line Search $[*]$;
            Set $\mathbf{x}_{k+1}^{[p]} \leftarrow \mathbf{x}_k^{[p]} + \alpha_k^{[p]} \mathbf{s}_k^{[p]}$;
            Set $k \leftarrow k + 1$;
        end
        Return $\mathbf{U}^{[p]} \leftarrow \zeta^{-1}\left(\mathbf{x}_k^{[p]}\right)$ $[*]$;
    end
end
```

3.5 Results

The case study that has been developed throughout this work lends itself to two levels of analysis in the postprocessing phase: a physical one (impact of friction damping on the frequency response) and a computational one (benefits of parallel computing in terms of solver time). In the following, we analyze a few notable results covering this topics.

The central engineering goal that motivated the development of friction-damped devices is represented by the possibility to manipulate the frequency response both in terms of position of the resonance along the spectrum and amplitude of the vibrations, things we achieve here too, as proven in Table 3.1 and Figure 3.1. Depending on the specific requirements of the system, the designers can optimize the normal pre-load N_0 to meet a trade-off between minimum peak amplitude and the furthest possible shift in the resonance frequency. Figure 3.1, in particular, proves the two-fold effect of adding a point of contact: the energy dissipated through friction causes the lower amplitude at resonance whilst the increasing normal load tends to transform this contact point effectively into a new constraint when it forces a fully sticking behaviour.

N_0	Norm. peak amplitude
0%	1.0000
1%	0.5524
2%	0.1481
6%	0.0534
20%	0.1301
40%	0.2416
60%	0.3526
100%	0.5306

Table 3.1. Values of the normalized peak amplitudes at various pre-loads, as percentage of minimum pre-load to reach a full stick response (see Table A.1).

As it was hinted at in Section 2.5, the number of harmonic components to be used in the Fourier series expansion depends on the problem in question: in the case of our 1D system, the first harmonic component carries with it most of the information of the frequency response, higher order *odd* harmonics are orders of magnitude smaller than the first one, while the *even* harmonics are only numerical noise, which is to be expected given the purely harmonic forcing term defined in Equation (2.27) (see Figure 3.2). Despite this, the collective impact on the results of higher order harmonics is clearly visible in the region of the spectrum where the nonlinear effects are stronger (see Figure 3.3). It can be seen how truncating the Fourier series expansions at the first harmonic gives a slight overestimation of the amplitude, while adding more and more terms tends to converge towards a lower value. In general, it is guaranteed that a periodic and *physically realizable* waveform (that is, with finite energy in one period) has a convergent Fourier series (see [15] on the convergence of Fourier series).

However, the key result of this work is represented in Figure 3.4 and in Table 3.2, which

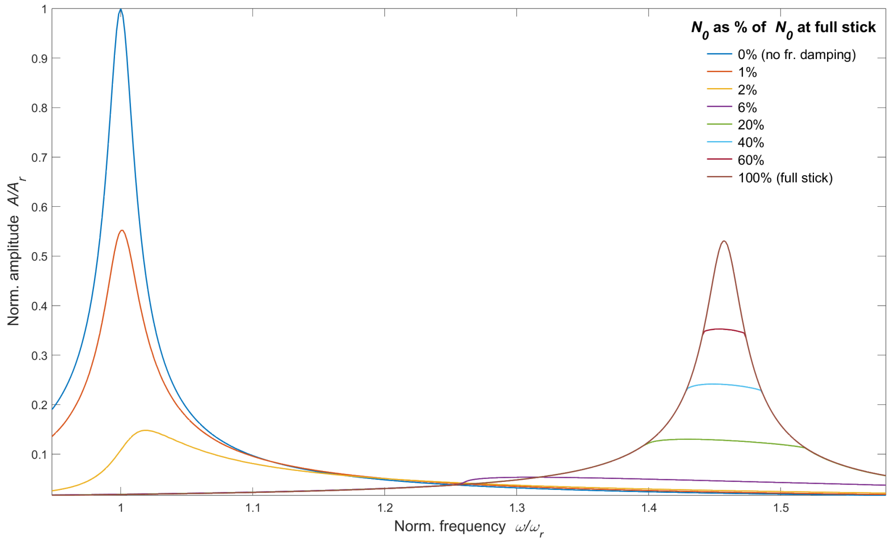


Figure 3.1. Frequency response of node with maximum vibration amplitude at various pre-loads (see Table A.1).

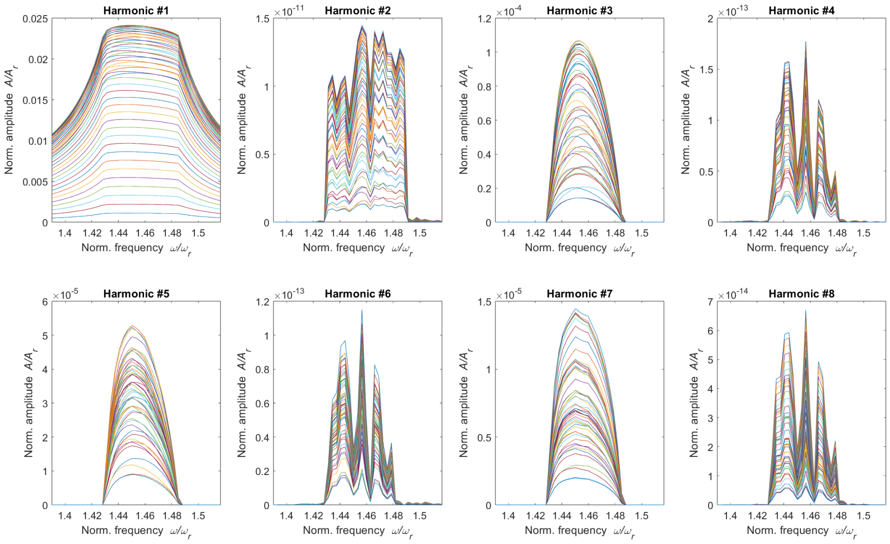


Figure 3.2. Response of 8 harmonic components, zoomed-in near the shifted resonance (for all 20 degrees of freedom in the grid, computed with $N_0 = 40\%$, see Table A.2).

prove that Algorithm 2 is effective at providing a significant speedup in computation time with good scalability above 1000 grid elements. As expected, for a lower number of degrees of freedom, instead, the communication overhead between cores renders ineffective

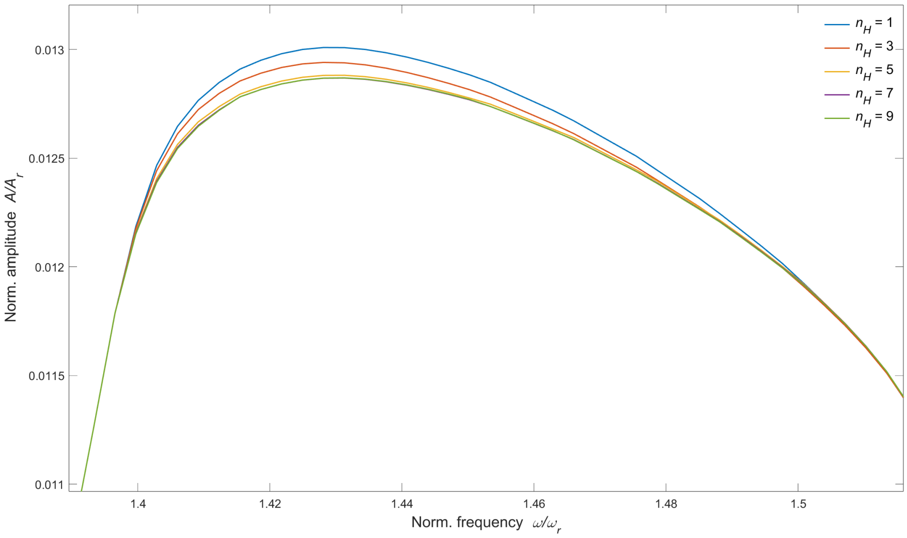


Figure 3.3. Behaviour of node with maximum vibration amplitude with increasing number of harmonics used, zoomed-in near the shifted resonance (computed with $N_0 = 20\%$, see Table A.2).

the faster solution of problems of smaller size.[□] Table 3.2, in particular, suggests that the combination of the algorithm, its implementation and the machine on which it was ran on, experiences something called *superlinear speedup*, with a reduction in time far superior than just a halving for the doubling of the number of cores. This phenomenon, although unusual since it breaches of theoretical limits of speedup, has been analyzed by some authors and the most plausible explanation given revolves around how different access speeds to the various *cache levels* of modern CPUs interact with operations of elevated complexity (see [18]).

As a procedural note, the time that MATLAB needs to allocate space when calling `parpool()` for a new pool of parallel workers to be created is not included in this analysis, being more akin to software startup time rather than computation time (for reference it was measured at $8 \div 16s$).

Another parameter that influences significantly the array size (as was shown in (3.25)) and, consequently solution time, is the number of harmonics n_H; this was also tested successfully, as shown in Figure 3.5. In this instance, the comparison of scalability does include serial code runs with the same physical parameters, performed with a mildly improved version of the code developed for the analysis in the article [11] and kindly provided by one of the authors who was, at the same time, supervisor of this thesis. The vastly superior efficiency of the parallel code which can be derived from Figure 3.5 is the reason

[□] The machine on which this was ran on uses AMD Ryzen 7 8845HS (3.80 GHz) CPU with 32 GB of RAM. MATLAB version: R2024a.

why serial code runs where not included in the more detailed performance analysis given in Figure 3.4.

N_{cores}	Solution Time (s)	Relative speedup
2	684.05	–
4	131.05	×5.2
8	31.31	×21.8
16	14.10	×48.5

Table 3.2. Values of the total solution time (in the case of 4000 elements, see Table A.3).

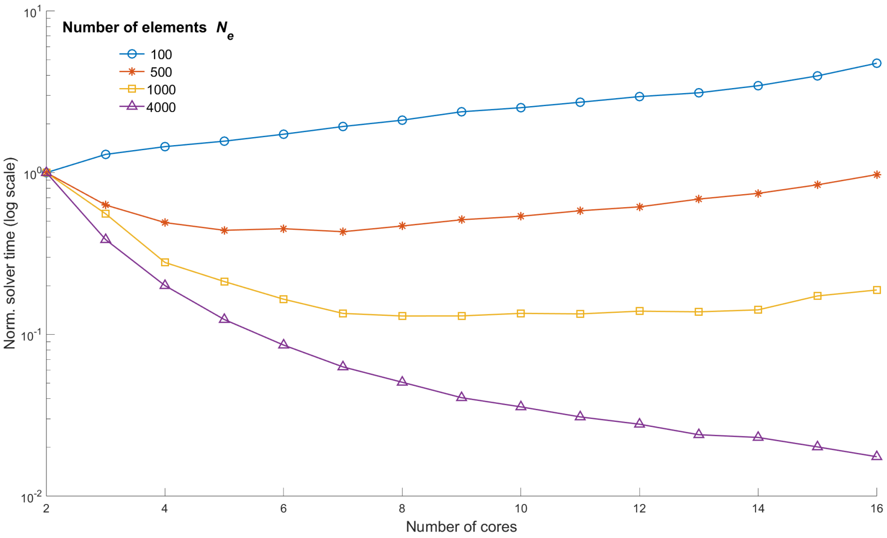


Figure 3.4. Behaviour of solution time (normalized with respect to the results for 2 cores) at various mesh densities (see Table A.3).

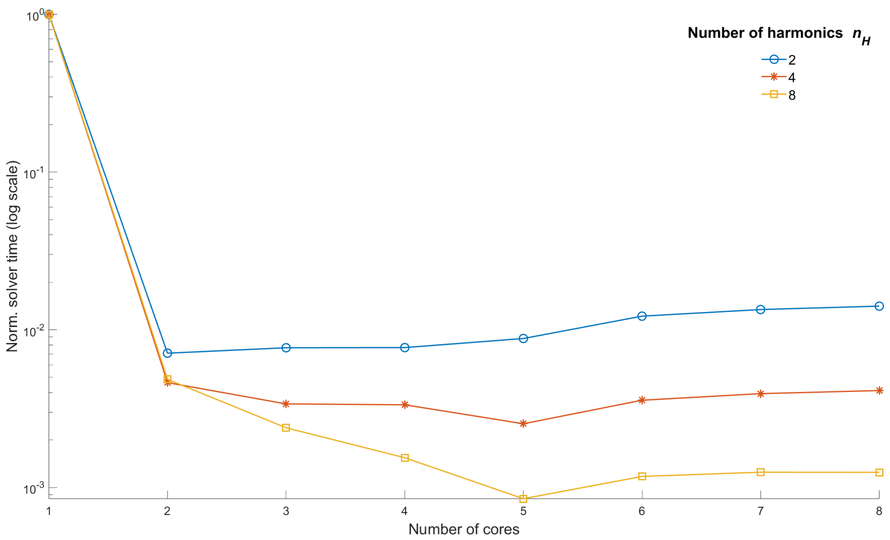


Figure 3.5. Behaviour of solution time (normalized with respect to the results for the serial code) with different numbers of harmonic components (see Table A.4).

Appendix A

Solution parameters

PHYSICAL PARAMETERS			NUMERICAL PARAMETERS		
Symbol	**Value**	**Description**	**Symbol**	**Value**	**Description**
E	2.0×10^5 MPa	Young's Modulus, steel	$\mathscr{F}$	$300 \div 500$ Hz	Frequency range
ρ	7.8×10^{-12} kg/mm^3	Mass density, steel	n_H	1	Number of harmonics
α_m	1.0×10^{-5}	Modal damping ratio	N_e	50	Number of elements
A	10 mm^2	Beam section area	N_s	4	Number of subdomains
L	4000 mm	Beam length	$\mathrm{max}_{\mathrm{it}}$	50	Newton max iterations
k_t	1.0×10^3 N/mm	Tan. contact stiffness	tol	1×10^{-20}	Convergence tolerance
μ	0.5	Friction coefficient	eps	1×10^{-8}	Finite difference step
N_0	[0 0.5 1 3 10 20 30 50] N	Normal pre-loads	α	1	Initial step length
k_n	1.0×10^3 N/mm	Nor. contact stiffness	τ	0.5	Line search ratio
	Beam point: L	Clamped end	c	0.1	Sufficient decrease
	Beam point: 0	Contact point	θ	0.2	α recovery rate
	Beam point: $L/2$	Load application	α_{lim}	0.001	α limit value

Table A.1. Summary of all parameters used for testing multiple normal pre-loads.

PHYSICAL PARAMETERS			NUMERICAL PARAMETERS		
Symbol	**Value**	**Description**	**Symbol**	**Value**	**Description**
E	2.0×10^5 MPa	Young's Modulus, steel	$\mathscr{F}$	$440 \div 480$ Hz	Frequency range
ρ	7.8×10^{-12} kg/mm^3	Mass density, steel	n_H	$1 \div 9$	Number of harmonics
α_m	1.0×10^{-5}	Modal damping ratio	N_e	50	Number of elements
A	10 mm^2	Beam section area	N_s	4	Number of subdomains
L	4000 mm	Beam length	$\max_{\text{it}}$	50	Newton max iterations
k_t	1.0×10^3 N/mm	Tan. contact stiffness	tol	1×10^{-20}	Convergence tolerance
μ	0.5	Friction coefficient	eps	1×10^{-8}	Finite difference step
N_0	[10 20] N	Normal pre-load	α	1	Initial step length
k_n	1.0×10^3 N/mm	Nor. contact stiffness	τ	0.5	Line search ratio
	Beam point: L	Clamped end	c	0.1	Sufficient decrease
	Beam point: 0	Contact point	θ	0.2	α recovery rate
	Beam point: $L/2$	Load application	α_{lim}	0.001	α limit value

Table A.2. Summary of all parameters used for testing multiple harmonics.

PHYSICAL PARAMETERS			NUMERICAL PARAMETERS		
Symbol	**Value**	**Description**	**Symbol**	**Value**	**Description**
E	2.0×10^5 MPa	Young's Modulus, steel	$\mathscr{F}$	$300 \div 500$ Hz	Frequency range
ρ	7.8×10^{-12} kg/mm^3	Mass density, steel	n_H	1	Number of harmonics
α_m	1.0×10^{-5}	Modal damping ratio	N_e	[100 500 1000 4000]	Number of elements
A	10 mm^2	Beam section area	N_s	$2 \div 16$	Number of subdomains
L	4000 mm	Beam length	$\max_{\text{it}}$	50	Newton max iterations
k_t	1.0×10^3 N/mm	Tan. contact stiffness	tol	1×10^{-20}	Convergence tolerance
μ	0.5	Friction coefficient	eps	1×10^{-8}	Finite difference step
N_0	20 N	Normal pre-load	α	1	Initial step length
k_n	1.0×10^3 N/mm	Nor. contact stiffness	τ	0.5	Line search ratio
	Beam point: L	Clamped end	c	0.1	Sufficient decrease
	Beam point: 0	Contact point	θ	0.2	α recovery rate
	Beam point: $L/2$	Load application	α_{lim}	0.001	α limit value

Table A.3. Summary of all parameters used for testing parallel speedup (1).

PHYSICAL PARAMETERS			NUMERICAL PARAMETERS		
Symbol	**Value**	**Description**	**Symbol**	**Value**	**Description**
E	2.0×10^5 MPa	Young's Modulus, steel	$\mathscr{F}$	$440 \div 480$ Hz	Frequency range
ρ	7.8×10^{-12} kg/mm^3	Mass density, steel	n_H	$[2\,4\,8]$	Number of harmonics
α_m	1.0×10^{-5}	Modal damping ratio	N_e	100	Number of elements
A	10 mm^2	Beam section area	N_s	$2 \div 8$	Number of subdomains
L	4000 mm	Beam length	$\max_{\text{it}}$	50	Newton max iterations
k_t	1.0×10^3 N/mm	Tan. contact stiffness	tol	1×10^{-20}	Convergence tolerance
μ	0.5	Friction coefficient	eps	1×10^{-8}	Finite difference step
N_0	20 N	Normal pre-load	α	1	Initial step length
k_n	1.0×10^3 N/mm	Nor. contact stiffness	τ	0.5	Line search ratio
	Beam point: L	Clamped end	c	0.1	Sufficient decrease
	Beam point: 0	Contact point	θ	0.2	α recovery rate
	Beam point: $L/2$	Load application	α_{lim}	0.001	α limit value

Table A.4. Summary of all parameters used for testing parallel speedup (2).

Bibliography

[1] Philip Avery and Charbel Farhat. "The FETI Family of Domain Decomposition Methods for Inequality-constrained Quadratic Programming: Application to Contact Problems with Conforming and Nonconforming Interfaces". In: *Computer Methods in Applied Mechanics and Engineering* 198 (2009). ISSN: 0045-7825. URL: https://doi.org/10.1016/j.cma.2008.12.014.

[2] Daniele Boffi, Franco Brezzi, and Michel Fortin. *Mixed Finite Element Methods and Applications*. Springer-Verlag, 2013. ISBN: 978-3-642-36518-8. URL: https://doi.org/10.1007/978-3-642-36519-5.

[3] Claudio Canuto. *Notes on Partial Differential Equations*. Lecture notes. July 2009.

[4] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and Applications of Finite Element Analysis, 4th Edition*. John Wiley & Sons, 2002. ISBN: 978-0-471-35605-9. URL: https://www.wiley.com/en-us/Concepts+and+Applications+of+Finite+Element+Analysis%2C+4th+Edition-p-9780471356059.

[5] Victorita Dolean, Pierre Jolivet, and Frédéric 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. URL: https://doi.org/10.1137/1.9781611974065.

[6] Zdeněk Dostál, Tomáš Kozubek, Marie Sadowská, and Vít Vondrák. *Scalable Algorithms for Contact Problems, Second Edition*. Springer Cham, 2023. ISBN: 978-3-031-33580-8. URL: https://doi.org/10.1007/978-3-031-33580-8.

[7] Zdeněk Dostál et al. "FETI Based Algorithms for Contact Problems: Scalability, Large Displacements and 3D Coulomb Friction". In: *Computer Methods in Applied Mechanics and Engineering* 194 (2005). ISSN: 0045-7825. URL: https://doi.org/10.1016/j.cma.2004.05.015.

[8] Charbel Farhat. "A Lagrange Multiplier Based Divide and Conquer Finite Element Algorithm". In: *Computing Systems in Engineering* 2 (1991). ISSN: 0956-0521. URL: https://doi.org/10.1016/0956-0521(91)90015-W.

[9] Charbel Farhat, Kendall Pierson, and Michel Lesoinne. "The Second Generation FETI Methods and their Application to Parallel Solution of Large-scale Linear and Geometrically Non-linear Structural Analysis Problems". In: *Computer Methods in Applied Mechanics and Engineering* 184 (2000). ISSN: 0045-7825. URL: https://doi.org/10.1016/S0045-7825(99)00234-0.

[10] Charbel Farhat and Francois-Xavier Roux. “A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm”. In: *International Journal for numerical methods in engineering* 32 (Oct. 1991). ISSN: 0029-5981. URL: https://doi.org/10.1002/nme.1620320604.

[11] Christian Maria Firrone and Stefano Zucca. “Modelling Friction Contacts in Structural Dynamics and its Application to Turbine Bladed Disks”. In: *Numerical Analysis - Theory and Applications* (Sept. 2011). ISSN: 978-953-307-389-7. URL: http://dx.doi.org/10.5772/25128.

[12] Ilias S. Kotsireas, Panos M. Pardalos, Alexander Semenov, William T. Trevena, and Michael N. Vrahatis. “Survey of Methods for Solving Systems of Nonlinear Equations, Part I: Root-finding Approaches”. In: (Aug. 2022). URL: https://arxiv.org/abs/2208.08530.

[13] Malte Krack and Johann Gross. *Harmonic Balance for Nonlinear Vibration Problems*. Springer Cham, 2019. ISBN: 978-3-030-14023-6. URL: https://doi.org/10.1007/978-3-030-14023-6.

[14] Frédéric Magoulès and François-Xavier Roux. “Lagrangian Formulation of Domain Decomposition Methods: A Unified Theory”. In: *Applied Mathematical Modelling* 30 (2006). ISSN: 0307-904X. URL: https://doi.org/10.1016/J.APM.2005.06.016.

[15] Norman Morrison. *Introduction to Fourier Analysis*. John Wiley & Sons, 1994. ISBN: 978-0-471-01737-0. URL: https://www.wiley.com/en-us/Introduction+to+Fourier+Analysis-p-9780471017370.

[16] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization, 2nd Edition*. Springer New York, 2006. ISBN: 978-0-387-30303-1. URL: https://doi.org/10.1007/978-0-387-40065-5.

[17] Alfio Quarteroni. *Modellistica numerica per problemi differenziali, 4a edizione*. Springer - Verlag Italia, 2008. ISBN: 978-88-470-0841-0. URL: https://doi.org/10.1007/978-88-470-5782-1.

[18] Sasko Ristov, Radu Prodan, Marjan Gusev, and Karolj Skala. “Superlinear Speedup in HPC Systems: why and when?” In: *Annals of Computer Science and Information Systems* 8 (2016). ISSN: 2300-5963. URL: https://doi.org/10.15439/2016F498.

[19] Christian Siewert, Lars Panning, Jörg Wallaschek, and Christoph Richter. “Multiharmonic Forced Response Analysis of a Turbine Blading Coupled by Nonlinear Contact Forces”. In: *Proceedings of the ASME Turbo Expo 2009: Power for Land, Sea, and Air* Volume 6: Structures and Dynamics, Parts A and B (June 2009). URL: https://doi.org/10.1115/GT2009-59201.

[20] Stefano Zucca. *Dinamica dei rotori per applicazioni aerospaziali*. Lecture notes. 2021.