A novel framework for the Bispectral Analysis of large structures subjected to Non-Gaussian wind loading

by

Michele Esposito Marzino

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Engineering Sciences

Sept 2024

Members of the Jury

President	Prof. Laurent DUCHENE University of Liège
Advisor	Prof. Vincent DENOËL University of Liège
	Prof. Romain BOMAN University of Liège
	Dr. Marco Lucio CERQUAGLIA Bureau d'Étude Greisch
	Prof. Christophe GEUZAINE University of Liège
	Prof. Massimiliano GIOFFRÈ University of Perugia
	Prof. Michael SHIELDS Johns Hopkins University

To the forgotten ones.

Contents

Abstract

The non-Gaussian nature of turbulent wind loading has been long accepted. Nevertheless, although this might have significant influence on design quantities, it seems to be very often ignored when it comes to practical applications, specially when dealing with real civil engineering structures, ranging from medium to quite large dimensions. Unfortunately, the need for significantly high computational power has been the main challenge to overcome. However, things have evolved quite fast in that domain over the last few years, possibly providing better tools for its effective computation.

Proper Orthogonal Decomposition (POD) and Reduced Order Model (ROM) techniques have also helped scientists and practitioners, in several fields including Wind Engineering, to tackle the complexities of their problems. Using physical and mathematical tools, POD can help at reducing the problem dimensionality, and making it more affordable to be solved numerically, whenever analytical formulations are not derivable.

This work aims at presenting a new framework for tackling the Bispectral problem applied to large civil engineering structures, under non-Gaussian wind loads. To do so, mathematical and numerical tools will be employed. A novel formulation of application of POD techniques to the Bispectral problem will be provided. Also, a novel algorithmic arrangement for tackling efficiently the computation of non-Gaussian features of the wind load, i.e. the Bispectrum, will be presented.

Coupling these two aspects, application of Bispectral analysis to real civil engineering examples will be provided. This will lead to the final effort of providing a reasonable answer to a fundamental question: when exactly is a Bispectral analysis needed?

Acknowledgements

This Thesis was made possible thanks to the financial support of the Belgian SPW (Service Public Finances Wallonie) and Bureau d'Étude Greisch. In particular, Bureau d'Étude Greisch which allowed me to be part of the FINELG2020 Research Project (Convention Nb. 8096), without which this Thesis would certainly not exist. I am thereby thankful for the opportunity.

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

In that, I thank Prof. David Colignon from University of Liège, for providing me direct support as a local CÉCI ULiège system administrator, and together with Dr. Orian Louant, for the very interesting and fruitful discussions.

Also, I want to thank Thibaud Bastin, Research Engineer at Bureau d'Étude Greisch. Not only for the fruitful discussions that we could have during these years of close collaboration. But also for the patience that you have showed, to all my constant requests, and the support in providing me all the high-quality figures that are present in this Thesis in the applications part. Without your help and support, this Thesis would have not been of the same quality and clarity. It has been a pleasure working with you.

If someone would have told me, few years ago, that in some yeas from there I would have written a PhD thesis, I would have certainly laughed. Not because I did not admire or respect the idea, but because I felt it far from what I thought I wanted to strive for. Yet, here I am, writing these last, few, lines of it. And I can strongly admit, in the good and the bad, that I am glad I proved myself wrong. This journey has been something unique. Something that one could not easily emulate differently.

Vincent, your guidance throughout this journey has been more than a mere blinding light. Your passion, your excitement, and your encouragement have been to me (as I am sure it is for everyone of us having the chance to work on your side) as an inciting father to its children when motivating, supporting them in a constructive way, regardless. Your way of handling, and curing both the professional and the personal aspects of everything are to me a big lesson, that I will carefully guard from here onwards. Don't ever loose it. It is amongst the most precious things that I will take from this experience.

In that, I also want to thank my collegues, past and present. Kevin, having you as my office mate has been a real pleasure. Though this might sound weird to the many, I really enjoyed our multi-lingual conversations, with me speaking French

and you Italian. If this has certainly helped me improve my French, to me it has been a joy seeing you enjoying speaking my native language, and seeing you improving day after day. Your commitment in this has also been a further proof of the amazing person you are. I really wish you the best, that you might find your passion and give all the good that I know you are capable to give. Whenever you will come to Italy, not only you will be able to communicate as a quasi-native Italian speaker, but you will also always have a person more than happy to greet you there. For that, I could never thank you enough for all the oranges you brought me, as a way to remind, for an instant, of home.

Speaking of Italy, I recall of an ancient saying we have: "Chi trova un amico, trova un tesoro". Well, while a saying is a saying, a saying is also often true, if the metaphor is well interpreted. To Alex, Chathura and Tudor (in alphabetical order, otherwise you could shuffle them as you wish) I want to say: I found it! The only thing, I could not invite you on my yacht yet, for that you should give me the time to find the other, less important, treasure. These years here in Liège with you have been a journey, that we could easily write a book on it. Indeed, moments have not always been good and bright, light and happy. We also shared lots of less good moments. And I got wrong with you many times. Yet, I still find you by my side, cheering for me, unconditionally, even though this might mean for our paths to separate. And if this will ever come, know that only kilometers will be able to separate them. Not memories. Memories, they will never expire.

Also, a big thank to everyone with which we shared good moments, throughout these years. Pietro, you revived my passion in taking pictures. Having your lense is a constant reminder of those good times.

To my life-long friend, Giuseppe. You always supported every choice I have made, even though this has meant for me to leave from our home, when we were still young and still had plenty of time to share, before turning into the "adult" life. Which we know, we will never be adult enough. I told you this many times, but they will never be enough as a reminder. You will always be with me, in my deepest memories, wherever life might take me.

I should also mention Alessandro. We have grown together. But we also lost our ways, for a long time. Hopefully, we are back on track. Our friendship is something that taught me, and made me grow, a lot. I thank you for that.

We are approaching the end. Yet, there are still few, but fundamental persons, I need to say thanks.

To Alessia. If someone would have told me that a world-wide pandemic would have struck, keeping everyone locked in their homes (for those who had the luck to

have one.) for two years, but which would have given me the chance to meet my partner in crime, it would have been a fantasy story that even the most creative film producer would have struggled to imagine. Thinking about it, we'd better move to shoot one! To those who might not know the behind-the-scenes, I tell that it is mainly because of your unconditional love, support, cheering, healing, and everything that you've done, and still do, every day, if this Thesis has come to its end. For that, you deserve to be next to me as an author of it.

To my parents, Elisa and Amedeo. You have devoted your life since 28 years, to make mine better than it could have possibly been. To Amedeo, my Father. Thanks for having done the right choice, 27 years ago. And though our path has been rough, harsh at times, stating that I owe you my entire life would never and ever be enough to repay for all the good that you brought to me. I would love to tell these words to you, through my voice. But I know it would be broken. So I'll write them here. For you.

With this line, while placed at the beginnig of it, I put a final, ending point to this Thesis.

Nomenclature

α_{Ia}^{m} = condensed modal wind coefficients

$\boldsymbol{\upsilon}$ = matrix collecting all of wind components

$\ddot{\mathbf{q}}(t)$ = vector of modal accelerations

$\ddot{\mathbf{x}}(t)$ = vector of structural accelerations

$\dot{\mathbf{q}}(t)$ = vector of modal velocities

$\dot{\mathbf{x}}(t)$ = vector of structural velocities

γ = Euler's constant

γ_3 = skewness coefficient

γ_4 = kurtosis coefficient

γ_e = excess kurtosis coefficient

$\Gamma_{u_{IJ}}(\omega)$ = spatial coherence function of $u(t)$

$\mathcal{A}_{I}^{m}$ = vector of condensed modal wind coefficients

ω = circular natural frequency

$\overline{U}$ = meand wind speed

$\overrightarrow{V}$ = spatial wind velocity vector

ρ = air volumic mass

σ = standard deviation

g_{G} = Gaussian peak factor

g_{NG} = non-Gaussian peak factor

i = imaginary number

$i(t)$ = wind incidence angle

I_u, I_v, I_w = turbulence intensities

$K_2(\omega_1, \omega_2)$ = second-order Volterra kernel

m_k = k-th order statistical moment

$R_{xxx}(\tau_1, \tau_2)$ = Bi-correlation function

$R_{xy}(\tau)$ = Cross-correlation function

$R_x(\tau)$ = Auto-correlation function

$S_{xy}(\omega)$ = Cross Power Spectral Density function

$S_x(\omega)$ = Auto Power Spectral Density function

μ_k = set of elementary wind turbulent components

$\mathbf{\Phi}$ = Modal matrix

$\mathbf{A}^{\mu_k}$ = matrix of wind wind coefficients, for k-th elementary wind turbulent component

$\mathbf{B_f}(\omega_1, \omega_2)$ = 3D-tensor of bispectra of nodal loads

$\mathbf{B_p}(\omega_1, \omega_2)$ = 3D-tensor of bispectra of modal loads

$\mathbf{B_q}(\omega_1, \omega_2)$ = 3D-tensor of bispectra of modal responses

$\mathbf{B_x}(\omega_1, \omega_2)$ = 3D-tensor of bispectra of response

$\mathbf{B_v}(\omega_1, \omega_2)$ = tensor of bispectra of wind components

$\mathbf{C}$ = Damping matrix

$\mathbf{C}^\star$ = Modal damping matrix

$\mathbf{E}$ = transformation matrix of wind forces from aerodynamic to structural nodes

$\mathbf{F}(\omega)$ = vector of Fourier transforms of applied loads

$\mathbf{f}(t)$ = vector of applied forces

$\mathbf{f}_{se}(t)$ = vector of self-exciting forces

$\mathbf{H}(\omega)$ = nodal Frequency Response Function (FRF)

$\mathbf{H}^{\star}(\omega)$ = modal Frequency Response Function (FRF)

$\mathbf{K}$ = Stiffness matrix

$\mathbf{K}^{\star}$ = Modal stiffness matrix

$\mathbf{L}$ = matrix of wind turbulence scales

$\mathbf{M}$ = Mass matrix

$\mathbf{M}^{\star}$ = Modal mass matrix

$\mathbf{P}(\omega)$ = vector of Fourier transforms of modal loads

$\mathbf{p}(t)$ = vector of modal loads

$\mathbf{Q}(\omega)$ = vector of Fourier transforms of modal responses

$\mathbf{q}(t)$ = vector of modal responses

$\mathbf{R}_{\boldsymbol{v}}(\Delta t)$ = tensor of correlation functions of wind components

$\mathbf{R}_{\boldsymbol{v}}(\Delta t_1, \Delta t_2)$ = tensor of Bi-correlation functions of wind components

$\mathbf{S}_{\boldsymbol{v}}(\omega)$ = tensor of PSD functions of wind components

$\mathbf{T}$ = transformation matrix of wind forces at aerodynamic nodes to forces at all structural degrees-of-freedom. Combination of E and X

$\mathbf{X}$ = transformation matrix of wind forces at structural nodes to forces at all structural degrees-of-freedom

$\mathbf{X}(\omega)$ = vector of Fourier transforms of structural responses

$\mathbf{x}(t)$ = vector of structural displacements

$E[\cdot]$ = mathematical expectation operator

`NDEGW` = number of wind components resulting from the degree of the wind model transformation

`NDOFs` = number of structural degrees-of-freedom

`NM` = number of structural vibration modes

`NNa` = number of aerodynamically loaded nodes

`NNs` = number of structural nodes

`NN` = number of aerodynamically loaded nodes that match with structural ones

$C_D(t)$ = Drag force coefficient

$C_L(t)$ = Lift force coefficient

$C_M(t)$ = Pitching moment coefficient

$F_D(t)$ = Drag force

$F_L(t)$ = Lift force

$F_M(t)$ = Pitching moment

ABL = Atmospheric Boundary Layer

CCC = Complete Cubic Combination

CFD = Computational Fluid Dynamics

CQC = Complete Quadratic Combination

CRSC = Cubic Root of Sum of the Cubes

CSD = Cross Spectral Density

CWE = Computational Wind Engineering

GRS = Global Reference System

HPC = High-Performance Computing

ICA = Independent Component Analysis

KLD = Karhunen-Loève Decomposition

MDOF = Multi Degree-of-Freedom

PCA = Principal Component Analysis

POD = Proper Orthogonal Decomposition

PSD = Power Spectral Density

SDOF = Single Degree-of-Freedom

SPOD = Spectral Proper Orthogonal Decomposition

SRSS = Square Root of Sum of the Squares

SVD = Singular Value Decomposition

THPC = Tail-Head-Previous-Current

VIV = Vortex Induced Vibrations

WRS = Wind Reference System

While this list of notations tries to be as exhaustive as possible, specially in reporting those symbols related to the most important concepts introduced in this Work, some of them might be missing. Nonetheless, they all are also well explained throughout the manuscript, whenever first introduced. As a general rule of tumb, lowercase and capital bold letters and symbols are used to denote vectors and matrices, respectively, except where explicitly stated. Superscripts $[\cdot]^T$ and $[\cdot]^*$ denote transpose and complex conjugate transpose (Hermitian) operators. Overhead bars denote mean values. Overhead hats are used to distinguish grouping conventions in the matrix of wind components, at all structural loaded nodes.

Chapter 1

Introduction

1.1 Context

In Civil Engineering, when dealing with flexible structures, wind action is certainly among the most important factors that must be accounted for in the design and verification stages.

When it comes to the dynamic effects of wind loads on flexible systems, a wide range of meteorological phenomena can be identified, based on different classifications. This work will exclusively focus on synoptic winds. This type of wind corresponds to classical stormy winds in neutral atmospheric conditions [1]. The wind profile across the height of the atmospheric boundary layer assumes the typical log law profile on average, but also some significant fluctuations around this mean value. These fluctuations result from the turbulence of the flow.

A given type of wind might interact in several ways with a flexible structural system. In particular, in the scientific community, a usual classification criterion opposes *steady* and *unsteady* models. The former assumes that the turbulent flow is unaffected by the structural motion, so that any fluid-structure interaction can be safely neglected, while in the latter fluid-structure interactions such as flutter are accurately modeled and can lead to aeroelastic instabilities. This work is restricted to the former type of model, where aerodynamic loads are expressed as memoryless transformations of the wind velocity field and of the structural velocity. This will be extensively discussed in the following Chapters.

Buffeting refers to the turbulent component of synoptic wind flows over flexible systems [1], and *buffeting analysis* refers to the structural analysis that determines displacements, internal forces, and ground reactions of the structure subjected to these turbulent wind actions. While unsteady phenomena such as flutter or galloping are critical for safety reasons, buffeting analysis usually assumes linear structural behavior and steady loading models. The aim of this analysis is to verify that the structure doesn't undergo excessively large internal

forces or displacements during severe storms. It is also important to consider the impact of these phenomena on comfort and fatigue issues in the long-term within the structure's lifetime [2].

Another important aspect of the dynamic wind action is its random character: at any point in space and instant in time, it is impossible to predict the exact value of the wind flow, even knowing all the values that were measured in the past.

For a given dynamical system, buffeting analysis consists in quantifying the variability in the structural response (the output), given the uncertainty of the random wind flow (the input). This challenge can be tackled using two different, but theoretically equivalent, approaches: either with a probabilistic analysis, either with time domain simulations based on a collection of realizations of the stochastic input, an approach that is also known under the terminology *Monte Carlo simulation*.

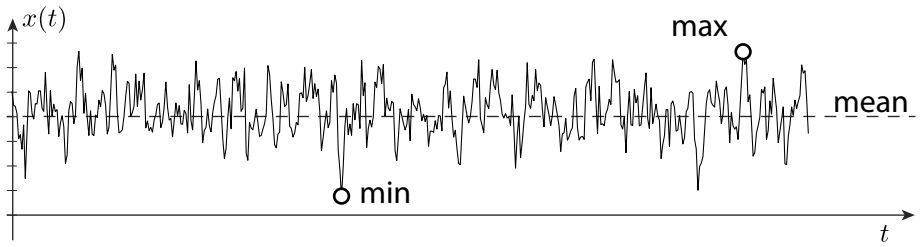


Figure 1.1: Example of a time series.

Time domain approach is likely to be chosen by practitioners. Indeed, a time domain approach is the most natural way of tackling a dynamical problem where loads, displacements and internal forces evolve with time. Time histories of wind loads and structural responses are easier to understand and visualise, see Figure 1.1. At any given time instant of the measuring window, both loading and structural responses take deterministic values, which therefore situates the problem in a comfortable deterministic setting. Deterministic is the opposite of random, meaning that a deterministic entity is known with certitude, since it comes from either measured or simulated values. With this approach, design quantities are the result of a statistical treatment of these time series; for instance the maximum displacement is simply obtained as the maximum value of the corresponding time series.

This approach has the limitation of requiring wind field samples as input to compute structural responses. Monte Carlo methods necessitate repeating time-domain analyses multiple times with different loading time histories to obtain statistically representative design quantities. Alternatively, a single, but very long time series can be used to determine reliable statistical estimates of the quantities of interest. In fact, a longer time series would more accurately encode

statistical information of the physical process, as a direct consequence of the Law of Large Numbers [3].

This concern was raised during the first half of the XX-th century by aerospace engineers, who were analyzing the buffeting of airplane wings subjected to wind turbulence and their resulting vibrations. They laid the foundations for the complementary probabilistic approach that were adopted by civil, structural, and bridge engineers in the early 1960s [1].

To overcome the limitations of Monte Carlo simulations, stochastic approaches were then developed. The fundamental question in these methods is: given the probabilistic description of the loading process, what is the resulting probabilistic description of the structural responses? In other words, how to propagate the stochastic input (wind loading) to the output of the structural system, without recoursing to samples of the wind field.

This approach has two main advantages:

- repeating the same analysis twice with wind loading characterized by the same statistical description yields identical responses (this property does not hold in a Monte Carlo approach).
- it also avoids the simulation of synthetic wind loading samples, or performing statistics of the long generated time series, which can be both tedious and expensive.

A probabilistic approach is an efficient way to solve problems without resorting to the statistical estimation required by the Monte Carlo method. For example, the probability of rolling a "6" on a standard six-sided die can be determined using a probabilistic approach by computing the ratio of favorable outcomes to the total number of equally probable results, yielding one out of six. In contrast, a Monte Carlo simulation involves simulating the experiment of rolling the die numerous times. The probability of obtaining a "6" is then statistically estimated by dividing the number of times a "6" is rolled by the total number of repetitions. This simple example demonstrates why avoiding Monte Carlo simulation is significantly faster.

Random processes like wind velocity or wind loads (input), and structural displacements and internal forces (output), are characterized by several statistical indicators. While a comprehensive description will be provided in Chapters 2 and 3, it is necessary to introduce some basic concepts here.

The mean and variance (square of standard deviation) represent the first two statistical moments, see Figure 1.2-a. For a Gaussian process, these are sufficient to fully describe the Probability Density Function (PDF). In non-Gaussian cases, additional statistical moments are necessary. The third and fourth moments quantify the statistical asymmetry and the flatness of the distribution

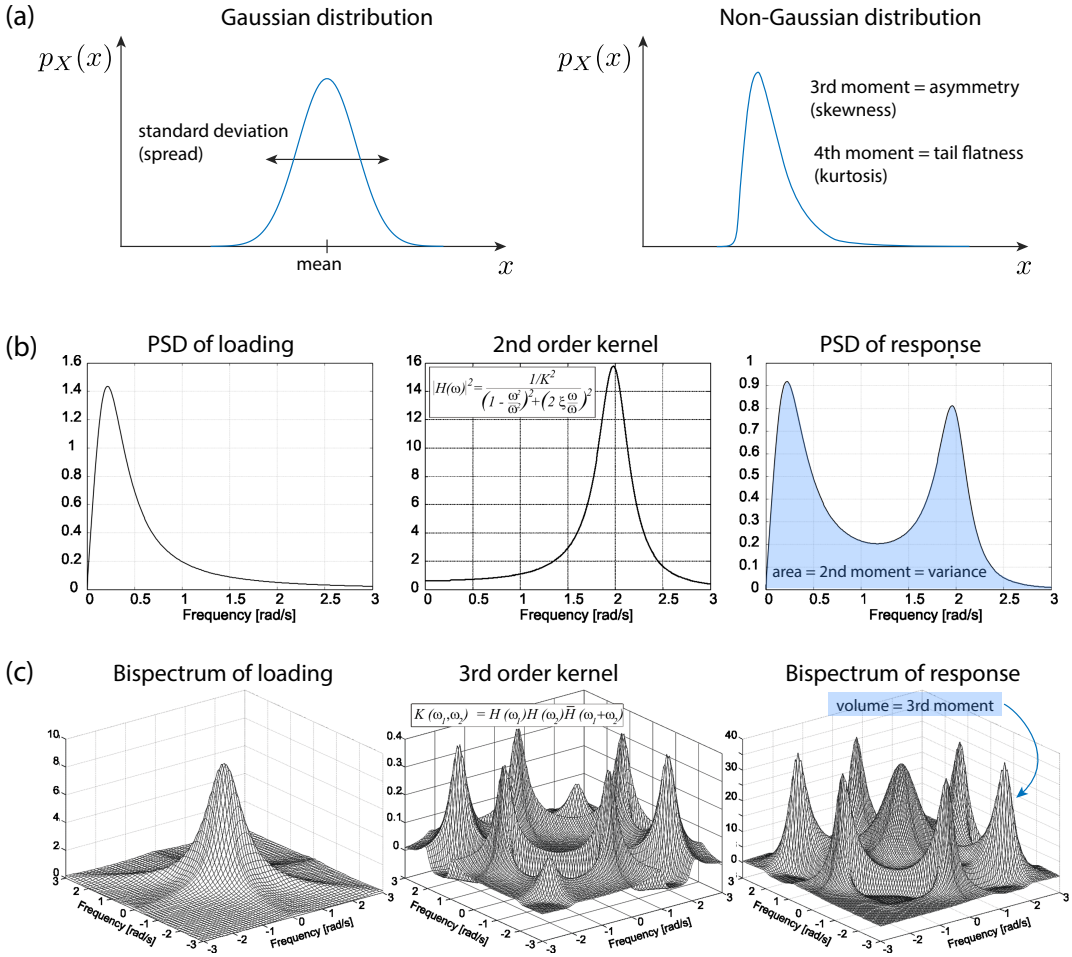


Figure 1.2: (a) illustration of the Gaussian distribution, and example of a non-Gaussian distribution, (b) Schematic of the working principle of a spectral analysis (2nd order), (c) Schematic of the working principle of a bispectral analysis (3rd order)

tails, respectively [4]. The first three or four moments are sufficient to represent a non-Gaussian process.

Furthermore, spectra of various orders are necessary to give an overview of the distribution of the moments in the frequency domain.

In a Gaussian context, the variance of a structural response is obtained by integrating the corresponding PSD, which is derived from the product of the input PSD and a structural kernel, see Figure 1.2-b. As detailed later, this second-order kernel is the squared norm of the frequency response function. Consequently, the PSD of the structural response shows contributions in the frequency ranges where the PSD of the loading is significant, as well as near natural frequencies, where the kernel exhibits sharp, narrow resonance peaks [5]. Capturing these sharp

peaks requires a fine mesh for the numerical integration of the response PSD.

Similarly, higher moments are obtained by integrating higher-order spectra, see Figure 1.2-c. The third moment of a structural response represents a crucial information for non-symmetrically distributed processes. It is obtained by integrating the bispectrum, which exhibits several resonance peaks resulting from the multiplication of the loading bispectrum by the third-order kernel. More details about the specific features of these kernels and higher order spectra will be given in the following Chapters.

This thesis mainly focuses on accurately integrating the third-order spectrum, alias the bispectrum, to determine the third statistical moment of the response. Increasing the order of the spectrum increases the problem's dimensionality. Unlike the PSD, that is distributed along the frequency axis, the bispectrum requires integration in a 2-D frequency space.

The complexity of determining higher-order spectra and the challenges associated with their numerical integration, particularly due to sharp resonance peaks, the probabilistic approach to non-Gaussian problems, such as buffeting analysis, has received limited attention in the past.

Practitioners typically follow one of two approaches: if the problem can be assumed to be Gaussian (both input and output), a simple probabilistic approach in the frequency domain is used. This approach is straightforward as the frequency space is only 1-D, and the variance of the response can be estimated by evaluating an integral, with special attention paid to sharp resonance peaks. With today's computational resources, this is no longer problematic. Alternatively, if the problem is non-Gaussian (input or output), the common solution strategy used today is a Monte Carlo simulation. This requires very long simulations, and often, the confidence bounds of the results are not provided, nor are the sensitivity and repeatability studies.

1.2 Motivation and objectives

In recent decades, various studies and scientists have begun to question the Gaussian assumption of wind loading, demonstrating that in many situations it cannot be considered as a Gaussian random process [6, 7, 8, 9, 10, 11].

In a Monte Carlo approach, abandoning the Gaussian assumption for the loading requires much longer simulated time histories. This is because the sample distributions of higher-order moments have larger standard errors.

Conversely, a stochastic approach requires evaluating higher-order spectra and integrating them to determine the higher-order moments. Establishing and integrating such spectra can be challenging in real-life applications due to numerical

difficulties in managing complex functions in higher-dimensional frequency spaces and the abstract nature of higher-order spectra concepts.

In this context, the main objective of this thesis is to provide a methodology for the efficient computation of higher-order stochastic spectra and corresponding moments. This work focuses on third-order statistics, specifically the integration of the bispectrum. While this problem has been addressed before [12], this thesis introduces a completely novel approach aimed at enabling bispectral analysis of large multi-degree-of-freedom (MDOF) structural systems subjected to non-Gaussian buffeting wind loads. The goal is to extend bispectral analysis capabilities to structures with several hundreds to thousands of degrees of freedom, surpassing the current limitation of about a dozen.

The mathematical derivations and proposed numerical solutions discussed in this work have a broader and more general scope. Although applied to third-order statistics in this work, the proposed structure can be conceptually extended and adapted to higher orders.

1.3 Manuscript structure

This thesis contains six chapters, structured as follows.

Chapter 2 introduces the Bispectral problem applied to a single-degree-of-freedom system.

Chapter 3 follows, providing a comprehensive overview and extension of concepts introduced in Chapter 2, as gradual introduction to the bispectral problem applied to large structures.

Chapter 4 begins with a brief overview of the nature of wind action, its main characteristics, and how they are considered in civil engineering applications. The main focus is on the random turbulent wind dynamic loading. A general overview of wind action is provided, followed by a discussion of current practices in civil engineering. This serves as a basis for understanding and comparing the generalized model of wind loading proposed in this thesis. A specialized version of this general model is then presented to illustrate the practical implications of different wind model choices on the resulting equations. The chapter concludes with the introduction of the general dynamic equations of motion and a brief discussion on extreme value theory. In this Chapter, the last Section is dedicated to the principles of the Proper Orthogonal Decomposition (POD) techniques, and how they are applied in Wind Engineering. Finally, its extension to Bispectral problems is formulated.

Chapter 5 explores the numerical developments that have been carried in this work. The first part focuses on discussing the problems intrinsic to a bispectral analysis under a numerical point of view. This is important to understand and gradually introduce the motivations that led to the conception and development

of the MESHER algorithmic arrangement, which is thoroughly detailed in the rest of the Chapter. In the last part, the algorithmic arrangement is discussed under a different point of view: its The last part discusses how the algorithmic arrangement is framed in the parallel computing framework: from current support and integration, to its potential and future implementations.

Finally, Chapter 6 presents some practical applications of Bispectral analysis. It discusses 3 applications: a first more academic example, useful for reproducibility, and two real applications, that aim at showing that bispectral analysis is not only of theoretical but also of practical interest, which might prove crucial in some circumstances.

The Thesis ends with Chapter 7, in which the findings are summarised and possible improvements and extensions of this work are discussed.

1.4 Personal contributions

The primary objective of this work is to develop a novel and effective solution for the bispectral problem, giving the opportunity to study large structures. Nonetheless, several other innovations were established along the way as part of the process of achieving the main goal.

In Chapter 4, two main contributions can be identified:

- a generalised formulation of wind forces model;
- a specialisation of such generalised model to a quadratic transformation, including non-Gaussian features of the wind load.

Moreover, in Section 4.8 the main contribution stands in the novel formulation of POD application to the non-Gaussian wind forces model. This extends current practice of leveraging POD in a Gaussian context, involving only PSDs of the wind loading.

The material presented in Chapter 5 is entirely a personal contribution, focusing on the algorithmic aspects that were conceived and developed from scratch. This includes the unified BSALIB library and its MESHER algorithm.

Finally, in Chapter 6, the main contribution lies in the derivation of the generalized “SK2BR-curve”, which was found to be an effective means of rapidly estimating the skewness of the structural response, given the background-to-resonant ratio of the structural response.

Chapter 2

Bispectral analysis of a single degree-of-freedom structure

This chapter focuses on introducing the bispectral problem theoretically, with consideration of its practical applications. It will cover in detail a straightforward, academic example: a single degree-of-freedom (SDOF) structure. The aim of this thesis is to develop the algorithmic framework that will allow extending the concepts presented here to multi degree-of-freedom (MDOF) structures.

2.1 A simple introductory problem

The equation of motion of an SDOF system reads:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t) \tag{2.1}$$

where m, c, k represent the structural characteristics (mass, damping, stiffness), $f(t)$ the applied load, $x(t)$ the dynamic structural response. This system could for instance represent the response in a single structural mode of vibration, a well-known concept in structural engineering, which is however recalled in Chapter 3.

For now, it is assumed that the wind force acting on this SDOF system is given by:

$$f(t) = \frac{1}{2}\rho c_D \Omega(\overline{U} + u(t))^2 \tag{2.2}$$

where $\rho = 1.225\ \frac{\text{kg}}{\text{m}^3}$ is the air density, c_D is a aerodynamic drag coefficient, Ω is a tributary loaded area, $\overline{U}$ is the mean wind speed, and $u(t)$ is the fluctuating component of the wind velocity, resulting from turbulence.

Being a result of turbulence, the longitudinal fluctuation of wind velocity $u(t)$ is precisely a random process. It is characterized in a probabilistic manner by

means of its variance and Power Spectral Density (PSD). As a result, the structural responses of interest, for instance the displacement $x(t)$ or the internal forces $kx(t)$, are also random processes. The structural analysis consists in determining them from a probabilistic point of view.

2.2 Statistics and Probability

Basic concepts related to the statistical treatment of random variables and random processes are introduced in this section. These concepts will be extensively used in later sections. The material presented here is available in many textbooks, see e.g. [13]. Major concepts are summarized to introduce them with the same notations as used later in this document. Advanced concepts related to MDOF systems will be developed and introduced in more detail in Chapter 3.

A **random variable** is the scalar outcome of a random event. Rolling a die or spinning a roulette wheel in a casino are typical examples of random events that provide scalar outcomes, such as a number from 1 to 6 or a color (black or red). Another example is the wind velocity at a given time and location, where the scalar outcome is a continuous variable, unlike the discrete variables in the previous examples.

A **random process** is a *sequence* of random variables with some neighbor-to-neighbor correlation, allowing for discussions of smoothness and differentiability in the continuous case. Simply put, a random process can be viewed as a random function, similar to a time signal [14].

2.2.1 Random variables

A random variable is defined on a set $\mathcal{S}$ of possible experimental outcomes of a random event [4]. When this set is discrete with a finite, countable, number of items, for instance $\mathcal{S} = \{1, \cdots, 6\}$ when rolling a die, it is possible to associate a probability of occurence with each possible outcome:

$$P_X(x_i) \qquad \forall x_i \in \mathcal{S} \tag{2.3}$$

where P_X represents the *probability function* of the random variable X. It is also convenient to deal with the *Cumulative Density Function* (CDF) F_X defined as:

$$F_X(x) = P(X \leq x) = \sum_{x_i \leq x} P_X(x_i) \tag{2.4}$$

for a discrete random variable. It quantifies the probability of the random variable to experience values lower than or equal to given threshold x. For example, the probability of rolling a die giving a result lower than 3.25 is 50%, knowing that

the favorable cases are 3 among 6 possible outcomes assuming that the die is not biased. Interestingly, the CDF is defined for $x \in \mathbb{R}$, while the probability function P_X is defined on $\mathcal{S}$ only.

For continuous random variables like those treated in this thesis, Equation (2.4) takes the form

$$p_X(x) = \frac{dF_X(x)}{dx} \quad ; \quad F_X(x) = \int_{-\infty}^{x} p_X(y)dy \tag{2.5}$$

where $p_X(x)$ refers to the *Probability Density Function* (PDF), and $p_X(x)dx$ represents the probability that the outcome X of the random event takes place in the interval $[x, x + dx]$.

Determining the PDF of a random variable from academic examples is relatively straightforward. However, when it comes to characterizing a real physical variable or process, it can be impractical to determine the complete PDF. Instead, mathematical moments such as the mean and variance can be used. In experimental testing, these moments are easier to determine than the entire PDF, which is often constructed from these moments while assuming a specific distribution.

2.2.1.1 Average and statistical moments

The *raw (statistical) moment* of order n of the continuous random variable X is defined as:

$$m_{n,X} = \mathbb{E}[X^n] = \int_{-\infty}^{+\infty} x^n p_X(x)dx \tag{2.6}$$

where $\mathbb{E}[\cdot]$ represents the *mathematical expectation operator*. The notation $m_{n,X}$ will be simplified to m_n in the following, when it does not lead to confusion.

The *average* or *mean value* of a continuous random variable X is the first statistical moment, $\mu_X = m_1$, or:

$$\mu_X = \mathbb{E}[X] = \int_{-\infty}^{+\infty} x p_X(x)dx. \tag{2.7}$$

Similarly, the *central (statistical) moment* of order n is defined as the moment of order n of the deviation of X with respect to its mean value μ_X :

$$\tilde{m}_{n,X} = \mathbb{E}[(X - \mu_X)^n] = \int_{-\infty}^{+\infty} (x - \mu_X)^n p_X(x)dx. \tag{2.8}$$

The notation $\tilde{m}_{n,X}$ will be simplified to $\tilde{m}_n$, when it is clear from the context that it refers to X.

2.2.1.2 Variance

The variance is the central moment of order 2, $n = 2$:

$$\tilde{m}_2 = \sigma_X^2 = \mathbb{E}[(X - \mu_X)^2] = \mathbb{E}[X^2] - \mu_X^2. \tag{2.9}$$

It represents the spread of the distribution around the mean value. It is easier to communicate with the *standard deviation* σ_X, defined as the square root of the variance. Indeed, the standard deviation has the same units as variable X.

If the random variable X is zero mean, $\mu_X = 0$, the second statistical moment m_2 and the variance σ_X^2 (second central moment) are equal. This can be generalised to any moment of order n.

2.2.1.3 Third moment and the skewness coefficient γ_3

In the context of this thesis, it is important to introduce the third central moment $\tilde{m}_3$:

$$\tilde{m}_3 = \mathbb{E}[(X - \mu_X)^3] = m_3 - 3m_2\mu_X + 2\mu_X^3. \tag{2.10}$$

As seen from Equation (2.8), the third central moment translates the asymmetry of the PDF around the mean value. This is a consequence of the oddness of the power n used to compute this moment. Indeed, occurrences of the random variable X beyond, and respectively below, the average μ_X contribute with different signs (positively, resp. negatively) to this odd-order moment.

Also, to provide graspable information, it is common practice to normalize the central moments with respect to the appropriate power of the standard deviation σ_X, so that dimensionless normalized central moments are defined for each order n:

$$\gamma_n = \frac{\tilde{m}_n}{\sigma_X^n}. \tag{2.11}$$

At third order, the normalized central moment, called *skewness coefficient*, is defined as

$$\gamma_3 = \frac{\tilde{m}_3}{\sigma_X^3}. \tag{2.12}$$

If the statistical distribution is symmetric around the mean value, $\gamma_3 = 0$. The reciprocal is not true.

2.2.1.4 Gaussian vs. non-Gaussian random variables

A Gaussian variable is a type of continuous random variable that is widely used in probability theory and statistics. The wind velocity at a given time and space

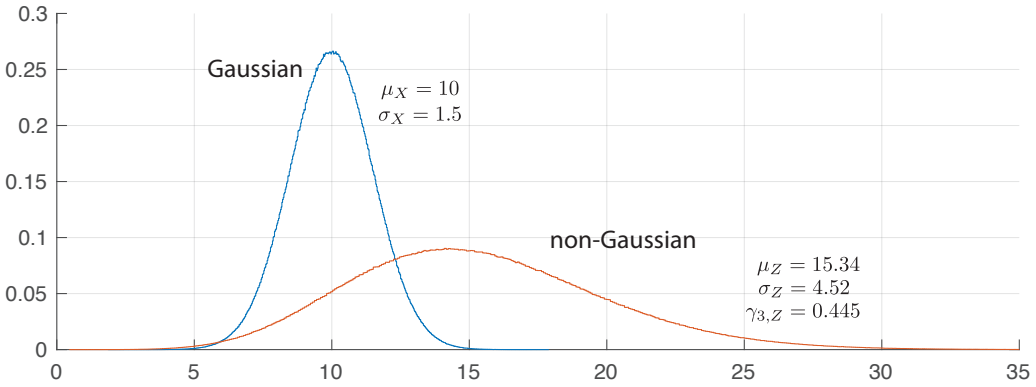


Figure 2.1: Example of distributions of Gaussian and non-Gaussian variables. For $\gamma_3 > 0$, the tail extends more to the right than to the left; and vice versa.

is typically modeled as a Gaussian variable. Its PDF is characterized by a bell-shaped curve, given by

$$p_X(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{2.13}$$

Substituting this expression in Equations (2.6) and (2.8) results in $m_1 = \mu$ and $\tilde{m}_2 = \sigma^2$, which show that μ actually represents the mean value of the distribution and σ represents the standard deviation. Furthermore, $\tilde{m}_3 = 0$ for the Gaussian variable, no matter the values of μ and σ.

The Gaussian distribution is important due to the Central Limit Theorem, which states that the sum of a large number of independent and identically distributed random variables tends to be normally distributed, regardless of the original distribution of the variables. This makes the Gaussian distribution a fundamental concept in statistics and many applied fields. It is characterized by only 2 parameters, μ and σ. It is also tempting to model a random variable as a Gaussian variable, when it is only known by its first two moments. This choice is supported by the maximum entropy principle [15].

Last but not least, the Gaussian distribution is always symmetrical, and therefore ceases to be appropriate as soon as the skewness of the random variable is non zero. For instance, squaring a Gaussian variable like $u(t)$, as in the right-hand side of (2.2), $(U + u(t))^2$, provides a non-Gaussian variable. It is asymmetrically distributed around the mean value. The first and simplest way to account for this non-Gaussianity is to represent the variable with its mean value, standard deviation, and skewness coefficient.

As an example, Figure 2.1 shows the distribution of X with $\mu_X = 10$ and $\sigma_X = 1.5$, which is typical of an average wind speed (10 m/s) and fluctuation at

a given time, as well as the distribution of $Z = 0.15X^2$ which could mimic the wind loading on the SDOF structure. By squaring the Gaussian wind velocity, the wind loading (considered at a given time) turns out to be a non-Gaussian random variable, with a skewness coefficient equal to $\gamma_Z = 0.445$. It is noticed that this value was obtained by simulation; it is possible to prove [5] that the skewness of the square of a Gaussian variable is asymptotically given by $3\sigma_X/\mu_X$ as $\sigma_X/\mu_X \to 0$, which would yield $\gamma_Z = 0.45$ in this case.

Several analytical methods exist to determine the entire PDF of the result Z of a transformation of a random process, such as $Z = 0.15X^2$. They are typically limited to low-dimensional transformation and struggle to be generalized with more complex transformations as those considered in the following chapters. The actual PDF of the quantities of interest are therefore usually not completely known. This is also the case for the structural responses, their randomness results from the randomness coming from the turbulence and its propagation into the dynamical systems. This is why current practice consists in estimating the first few moments and reconstructing a PDF based on them only.

For continuous variables defined on $\mathbb{R}$, when the first two moments only are available, the Gaussian assumption can be formulated. When the first three moments are known, 3-parameter distributions can be fitted, for instance the skew-normal distribution. Another well-known option to build a PDF from the first few moments is based on the Edgeworth expansion, which yields

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \mathrm{e}^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}} \left[1 + \sum_{n=3}^{\infty} \frac{\gamma_n}{n!} H_n\left(\frac{x-\mu_X}{\sigma_X}\right)\right]. \tag{2.14}$$

In this work, this series will be truncated after $n = 3$ since the statistics of the structural responses will be determined up to that order.

2.2.2 Random processes

Unlike a random variable, a random process is a random sequence resulting from an experiment whose outcome is not a scalar but a function, for instance of time, as in the cases treated in this thesis. Wind turbulence velocity is an example of a random process. At any given point in space, the random experiment of measuring the wind velocity with an anemometer would lead to a function of time, $U + u(t)$, hence a random process. Figure 2.2 shows an example of wind velocity where the average value U and the fluctuating component $u(t)$ are separated.

It is natural to imagine that, like for random variables, random process are best described by their *first-rank Probability Density Function* (PDF) $p_X(x_1, t_1)$, which is identical to the definition given for continuous random variables in Equation (2.5).

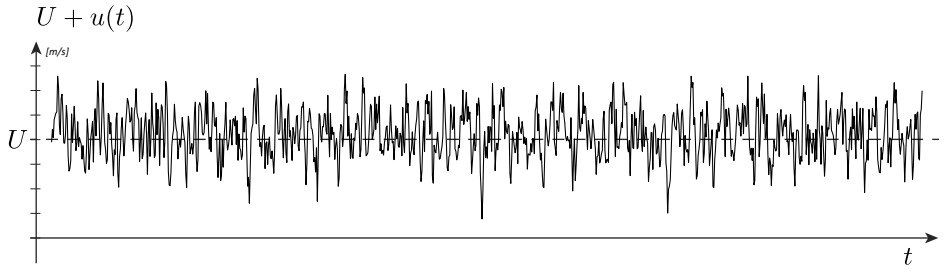


Figure 2.2: Example of wind turbulence velocity.

The quantity $p_X(x_1, t_1)dx_1$ represents the probability of finding the random variable $X(t_1)$ in the interval $[x_1, x_1 + dx]$. This is a natural extension of the definition given for a random variable since the random process $X(t)$ becomes a random variable as soon as time t is fixed. The first-rank probability density function therefore describes the random process as a collection of random variables [13]. However, the first-rank PDF does not provide information about the interdependence of values that can be taken by the random function at different times. To address this, it is necessary to define higher-rank PDFs. In the most general case, the *rank-n Probability Density Function* reads:

$$p_X(x_1, t_1; \ldots; x_n, t_n). \tag{2.15}$$

It is a joint PDF, and $p_X(x_1, t_1; \ldots; x_n, t_n)dx_1 \cdots dx_n$ represents the probability of jointly finding the random process X in the intervals $[x_i, x_i + dx_i]$, respectively for each of the time instants t_i $(i = 1, \cdots, n)$.

2.2.3 Moment functions of random processes

As for a random variable, statistical moments can be computed for a random process as well. However, as per their fundamental definition, when computing statistical moments of a random process, the result is also a function and not a scalar. For this reason, they are referred to as *moment functions*.

2.2.3.1 1st order: mean function

The first order moment function of a random process is the *mean function*, following the definition of mean value of a random variable given in Equation (2.7). As per the definition of a random process, in the most general case, the mean function depends on time t_1:

$$\mu_X(t_1) = \mathbb{E}[X(t_1)] = \int_{-\infty}^{+\infty} x p_X(x, t_1)dx \tag{2.16}$$

2.2.3.2 2nd order: autocorrelation function

The second order moment function of a random process is the *autocorrelation function*. As the name suggests, the autocorrelation function quantifies the correlation between values taken by a random process (and itself) at two different time instants. Mathematically, it reads:

$$R_{xx}(t_1, t_2) = \mathbb{E}[X(t_1)X(t_2)] = \int_{-\infty}^{+\infty} x_1 x_2 p_X(x_1, t_1; x_2, t_2) dx_1 dx_2 \tag{2.17}$$

where $p_X(x_1, t_1; x_2, t_2)$ is the second-rank joint probability density function.

If the two time instants in Equation (2.17) are identical, $t_1 = t_2$, one obtains the *mean-square value* at time t [13]:

$$R_{xx}(t_1, t_1) = \mathbb{E}[X^2(t_1)] \tag{2.18}$$

which resembles the definition of the second order moment of a random variable, given in Equation (2.9), with the exception here that it might depend on time.

Gaussian stochastic processes are characterized by a first-rank PDF following the Gaussian distribution given in Equation (2.13). They are therefore completely characterized by their mean value and standard deviation, possibly varying with time, which are related to $\mu_X(t_1)$ and $\mathbb{E}[X^2(t_1)]$. In stochastic design and modeling, Gaussian processes play a central role because of this reason, and the possibility to limit considerations at second rank, i.e. considering correlation of values taken by the process at two different times.

2.2.3.3 3rd order: bicorrelation function

As soon as non-Gaussianity is considered in a mathematical modeling, third order statistics at least need to be quantified. This requires the estimation of the third-rank PDF which translates the interdependence of the values taken by the process at three different times. As seen next, this will generalize the typical spectral analysis for linear systems to higher order spectral analysis [16, 4, 17].

Since the main focus of this thesis is on the influence of non-Gaussian wind loading on structures, it is necessary to define and make use of statistical moment functions higher than second. However, due to mathematical and numerical complexity limits, which will be extensively discussed in the following chapters, only moment functions up to third order will be treated, leaving higher orders to future studies.

The third order moment function of a random process is defined as

$$\begin{aligned} R_{xxx}(t_1, t_2, t_3) &= \mathbb{E}[X(t_1)X(t_2)X(t_3)] \\ &= \int_{-\infty}^{+\infty} x_1 x_2 x_3 p_X(x_1, t_1; x_2, t_2; x_3, t_3) dx_1 dx_2 dx_3 \end{aligned} \tag{2.19}$$

It is called the *bicorrelation function*, and defines the correlation of values taken by the random process at three different time instants. Considering the three time instants to be equal, $t_1 = t_2 = t_3$, the former equation specializes to the third statistical moment:

$$R_{xxx}(t, t, t) = \int_{-\infty}^{+\infty} x^3 p_X(x, t) dx = \mathbb{E}[X^3(t)]. \tag{2.20}$$

Together with the average and the variance, this moment can be used to obtain $\tilde{m}_3$ and reproduce a non-Gaussian PDF, based on the known values of the first few statistical moments, see Equation (2.14).

2.2.4 Stationary random processes

There are many kinds of random processes. In the context of this thesis, only stationary and ergodic processes will be considered, as they correspond well to synoptic winds.

A random process is said to be *strongly stationary* or *stationary in the strict sense* if its probability description is not affected by a time shift of time origin. This means that [13]

$$p_X(x_1, t_1; \ldots; x_n, t_n) = p_X(x_1, t_1 + a; \ldots; x_n, t_n + a) \tag{2.21}$$

no matter the value of $a \in \mathbb{R}$.

As a consequence, the first order density becomes independent of the time variable (see Equation (2.22)), while all higher densities depend on differences between the time arguments only. Mathematically, at first order, if the property is valid no matter a, a could be chosen as $a = -t_1$, so that

$$p_X(x, t_1) = p_X(x, t_1 + a) \quad \forall a \in \mathbb{R} \quad \rightarrow \quad p_X(x, t_1) = p_X(x, 0) \equiv p_X(x). \tag{2.22}$$

At second order, a could be chosen as $a = -t_1$ again, so that

$$p_X(x_1, t_1; x_2, t_2) = p_X(x_1, t_1 + a; x_2, t_2 + a) = p_X(x_1, 0; x_2, t_2 - t_1) \equiv p_X(x_1, x_2, \tau) \tag{2.23}$$

where $\tau = t_2 - t_1$ is the time lag between the two time arguments.

From Equations (2.22) and (2.23) it is straightforward to derive that the mean of a stationary process is time independent

$$\mu_X(t) = \mathbb{E}[X(t)] = \int_{-\infty}^{+\infty} x p_X(x,t)dx = \int_{-\infty}^{+\infty} x p_X(x)dx = \mu_X \tag{2.24}$$

and that the autocorrelation function (see Equation (2.17)) will be a function of the time lag $\tau = t_2 - t_1$ only:

$$\begin{aligned} R_{xx}(t_1,t_2) = \mathbb{E}[X(t_1)X(t_2)] &= \int_{-\infty}^{+\infty} x_1 x_2 p_X(x_1,t_1;x_2,t_2)dx_1dx_2 \\ &= \int_{-\infty}^{+\infty} x_1 x_2 p_X(x_1;x_2,\tau)dx_1dx_2 = R_{xx}(\tau). \end{aligned} \tag{2.25}$$

At third order, following similar arguments (choose $a = -t_1$), the third-rank probability density function is just a function of two time lags,

$$p_X(x_1,t_1;x_2,t_2;x_3,t_3) = p_X(x_1,t_1+a;x_2,t_2+a;x_3,t_3+a) \equiv p_X(x_1;x_2,\tau_1;x_3,\tau_2). \tag{2.26}$$

As a consequence, the bicorrelation function of a stationary random process is expressed as a function of two time lags $\tau_1 = t_2 - t_1$ and $\tau_2 = t_3 - t_1$:

$$R_{xxx}(\tau_1,\tau_2) = \mathbb{E}[X(t_1)X(t_2)X(t_3)] = \int_{-\infty}^{\infty} x_1 x_2 x_3 p_X(x_1;x_2,\tau_1;x_3,\tau_2)dx_1dx_2dx_3. \tag{2.27}$$

2.2.5 Ergodic random processes

The Ergodicity theorem allows determining the statistics of a random process $X(t)$ from a single, infinitely long, sample of the process. This theorem states that, for an ergodic process, every *ensemble average* can be replaced by a *time average along a single sample* [13]. Let $\hat{x}(t)$ be a sample of a given stationary random process $X(t)$. As such, one can make use of statistical treatment of signals tools and compute the mean

$$\hat{\mu} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \hat{x}(t)dt, \tag{2.28}$$

and the correlation integral [13]

$$\hat{R}_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \hat{x}(t)\hat{x}(t+\tau)dt, \tag{2.29}$$

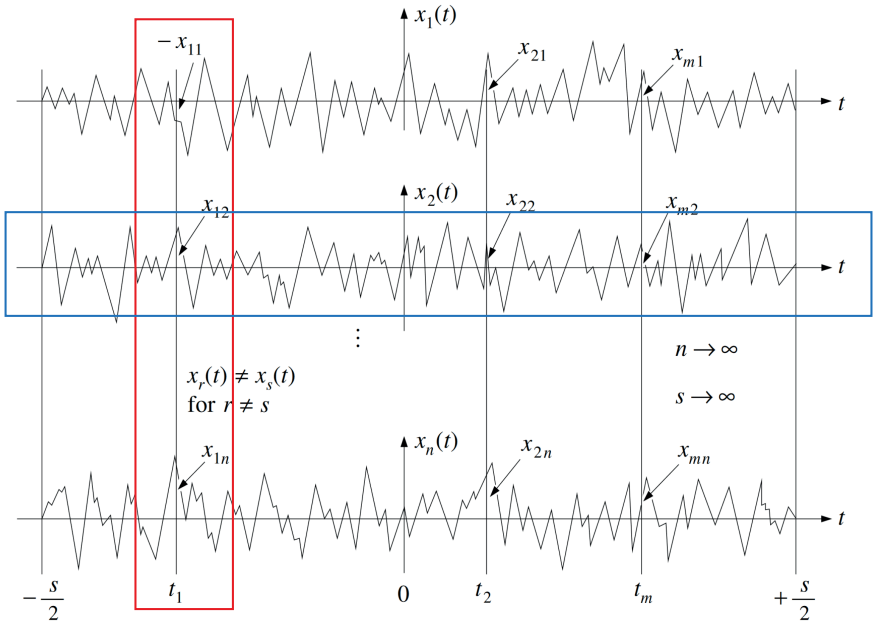


Figure 2.3: Graphical representation of an ergodic random process. Ensemble mean (red box), sample mean (blue box). Base figure taken from [14].

and the bicorrelation integral

$$\hat{R}_{xxx}(\tau_1, \tau_2) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \hat{x}(t)\hat{x}(t+\tau_1)\hat{x}(t+\tau_2)dt. \tag{2.30}$$

Then, the *ergodicity* property implies that

$$\hat{\mu} = \mu_X = \mathbb{E}[X(t)] \tag{2.31}$$

$$\hat{R}_{xx}(\tau) = R_{xx}(\tau) = \mathbb{E}[X(t)X(t+\tau)] \tag{2.32}$$

$$\hat{R}_{xxx}(\tau_1, \tau_2) = R_{xxx}(\tau_1, \tau_2) = \mathbb{E}[X(t)X(t+\tau_1)X(t+\tau_2)]. \tag{2.33}$$

Figure 2.3 shows a graphical representation of the ergodicity property. It shows a collection of samples of a random process, which evolve in time along the horizontal axis. The ensemble mean is obtained by averaging values in the vertical direction, following a cut along the whole set of realizations, at a given time instant t_1 (red box in Figure 2.3). If the random process is ergodic, this ensemble mean can be obtained from the expectation of any realizations of the random process (blue box in Figure 2.3).

One important aspect to remark is that an ergodic process is also stationary while a stationary process might not be ergodic. A practical implication of this

Theorem relates to the fact that long time series used in Monte Carlo simulations result in the same information as those provided by a probabilistic analysis.

2.2.6 The Power Spectral Density function

Let $X(t)$ be a stationary, ergodic random process. The *Power Spectral Density* (PSD) $S_x(\omega)$ is defined as the Fourier transform of the autocorrelation function:

$$S_{xx}(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} R_{xx}(\tau)e^{-i\omega\tau}d\tau. \tag{2.34}$$

As a consequence, the autocorrelation function is given by the inverse Fourier transform of the PSD, such that they form a Fourier pair:

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx}(\omega)\, e^{i\omega\tau}d\omega. \tag{2.35}$$

Evaluating the autocorrelation function of a stationary random process for zero time lag ($\tau = 0$), i.e. identical time instants $t_1 = t_2$, returns the second order moment, see Equation (2.18). If the random process is ergodic, this can be translated in

$$R_{xx}(\tau = 0) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x^2(t)dt = m_{2,x}. \tag{2.36}$$

It follows, from Equations (2.36) and (2.35), that the integral of the Power Spectral Density function of a random, stationary, ergodic process gives the second order statistical moment:

$$m_{2,x} = R_{xx}(\tau = 0) = \int_{-\infty}^{\infty} S_{xx}(\omega)\, d\omega. \tag{2.37}$$

For this reason, the PSD describes the distribution of the variance of the process in the frequency domain [13].

It is possible to prove that

$$S_{xx}(\omega) = \lim_{T\to\infty}\frac{2\pi}{T}\left|\hat{X}_T(\omega)\right|^2 = \lim_{T\to\infty}\frac{2\pi}{T}\hat{X}_T(\omega)\,\overline{\hat{X}}_T(\omega), \tag{2.38}$$

that is, the PSD is related to the Fourier transform of a long realization of a real ergodic process, defined as

$$\hat{X}_T(\omega) = \frac{1}{2\pi}\int_{-T/2}^{T/2}\hat{x}(t)\, e^{-i\omega t}dt. \tag{2.39}$$

Indeed, substitution of this expression in the previous one yields

$$\begin{aligned} S_{xx}(\omega) &= \lim_{T\to\infty} \frac{1}{2\pi T} \int_{-T/2}^{T/2} \hat{x}(t_1)\, e^{-i\omega t_1} dt_1 \int_{-T/2}^{T/2} \hat{x}(t_2)\, e^{i\omega t_2} dt_2 \\ &= \lim_{T\to\infty} \frac{1}{2\pi T} \iint_{-T/2}^{+T/2} \hat{x}(t_1)\, \hat{x}(t_2)\, e^{-i\omega(t_1-t_2)} dt_1 dt_2. \end{aligned} \tag{2.40}$$

Using the change of variable $\tau = t_1 - t_2$, and recalling the definition of the autocorrelation from long samples of an ergodic process, Equation (2.29), this yields

$$\begin{aligned} S_{xx}(\omega) &= \lim_{T\to\infty} \frac{1}{2\pi T} \iint_{-T/2}^{+T/2} \hat{x}(t_2+\tau)\, \hat{x}(t_2)\, e^{-i\omega\tau} d\tau dt_2 \\ &= \lim_{T\to\infty} \frac{1}{2\pi} \int_{-T/2}^{+T/2} R_{xx}(\tau)\, e^{-i\omega\tau} d\tau. \end{aligned} \tag{2.41}$$

After completion of the limit, assuming it exists, this is equivalent to the definition (2.34) and validates therefore the relation (2.38).

2.2.7 The Bispectrum

The *bispectrum* $B_x(\omega)$ of a stationary stochastic process $x(t)$ is defined as the twofold Fourier transform of the bicorrelation function:

$$B_{xx}(\omega_1, \omega_2) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} R_{xxx}(\tau_1, \tau_2) e^{-i\omega_1\tau_1} e^{-i\omega_2\tau_2} d\tau_1 d\tau_2. \tag{2.42}$$

As a consequence, the bicorrelation function is given by the twofold inverse Fourier transform of the bispectrum, such that they form a Fourier pair:

$$R_{xxx}(\tau_1, \tau_2) = \iint_{-\infty}^{\infty} B_x(\omega_1, \omega_2)\, e^{i\omega_1\tau_1} e^{i\omega_2\tau_2} d\omega_1 d\omega_2. \tag{2.43}$$

Evaluating the bicorrelation function of a stationary random process for zero time lags ($\tau_1 = \tau_2 = 0$), i.e. identical time instants $t_1 = t_2 = t_3$, returns the third order moment, see Equation (2.30). If the random process is ergodic, this can be

translated in

$$R_{xxx}(\tau_1 = 0, \tau_2 = 0) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x^3(t)dt = m_{3,x}. \tag{2.44}$$

The bispectrum is typically complex-valued, with the imaginary part often being non-zero [18]. It is for instance the case for local wind pressures measured on building facades [19]. Should these loads be used as an input in a subsequent structural analysis, it is important to include the imaginary part of the bispectrum in the analysis, which can also contribute to the response [19]. Bispectra are particularly useful in signal processing for analyzing non-Gaussian processes and understand sources of nonlinearity in dynamical systems. Indeed, since the bicorrelation encapsulates phase coherence between different frequency components, the bispectrum captures the interactions between pairs of frequencies (or their sum and difference) in the signal. These interactions inherently involve phase shifts between components. In particular, the imaginary part of the bispectrum arises due to the phase coupling between frequency components, which has become a standard tool to detect nonlinearities in signal processing [20, 21, 22]. In the context of this thesis, following the quasi-steady approach, the loading is considered as a polynomial transformation of a Gaussian process, and it is possible to show that is it therefore a real quantity [5]. Cross-bispectra however are complex-valued quantities but featuring symmetry properties such that their integral is actually a real quantity. This is expected since 3$^{\text{rd}}$ moments of real processes are real. In the following, focus will be on the real parts of the bispectrum and cross-bispectrum.

It follows, from Equations (2.43) and (2.44), that the integral of the bispectrum of a random, stationary, ergodic process gives the third order statistical moment:

$$m_{3,x} = R_{xxx}(\tau_1 = 0, \tau_2 = 0) = \iint_{-\infty}^{\infty} B_x(\omega)\, d\omega_1 d\omega_2. \tag{2.45}$$

For this reason, the bispectrum describes the distribution of the third statistical moment of the process in the 2-D frequency domain. Similarly to what was done at second order, it is possible to prove that

$$B_{xxx}(\omega_1, \omega_2) = \lim_{T\to\infty} \frac{4\pi^2}{T} \hat{X}_T(\omega_1)\, \hat{X}_T(\omega_2)\, \overline{\hat{X}}_T(\omega_1 + \omega_2) \tag{2.46}$$

that is, the bispectrum is related to the Fourier transform of a long realization of an ergodic process. Indeed, substitution of (2.39) in the previous one yields

$$B_{xxx}(\omega_1,\omega_2) = \lim_{T\to\infty}\frac{1}{2\pi T}\iiint_{-T/2}^{+T/2}\hat{x}(t_1)\,\hat{x}(t_2)\,\hat{x}(t_3)\,e^{-i\omega_1 t_1}e^{-i\omega_2 t_2}e^{i(\omega_1+\omega_2)t_3}dt_1dt_2dt_3 \tag{2.47}$$

Using the unit-Jacobian change of variable $\tau_1 = t_1 - t_3$ and $\tau_2 = t_2 - t_3$, and recalling the definition of the bicorrelation from long samples of an ergodic process, Equation (2.30), the previous expression becomes

$$\begin{aligned} B_{xxx}(\omega_1,\omega_2) &= \lim_{T\to\infty}\frac{1}{2\pi T}\iiint_{-\infty}^{+\infty}\hat{x}(t_3+\tau_1)\,\hat{x}(t_3+\tau_2)\,\hat{x}(t_3)\,e^{-i\omega_1\tau_1}e^{-i\omega_2\tau_2}d\tau_1d\tau_2dt_3 \\ &= \lim_{T\to\infty}\frac{1}{2\pi}\iint_{-\infty}^{+\infty}R_{xxx}(\tau_1,\tau_2)\,e^{-i\omega_1\tau_1}e^{-i\omega_2\tau_2}d\tau_1d\tau_2 \end{aligned} \tag{2.48}$$

which corresponds to the definition of the bispectrum, and validates therefore the relation (2.46).

These concepts can be extended to higher orders to compute, for instance, the tricorrelation and the trispectrum at fourth order. However, in this thesis, discussions and developments will be limited to third-order statistics only, i.e. bicorrelation and bispectrum at most, leaving higher orders for future studies.

2.2.8 System I/O relations

Let a deterministic causal Linear Time Invariant (LTI) dynamical system defined by its impulse response function $h(\tau)$, in such a way that the random output $y(t)$ of this system is obtained as a convolution of the random input $x(t)$ and the deterministic impulse response function:

$$y(t) = \int_{-\infty}^{t} h(\tau)x(t-\tau)d\tau. \tag{2.49}$$

The Fourier transform of this expression involves the Frequency Response Function (FRF) of this LTI system,

$$Y(\omega) = H(\omega)X(\omega) \tag{2.50}$$

where $H(\omega) = \int_{-\infty}^{\infty} h(\tau) e^{-i\omega\tau} d\tau$ is the Frequency Response Function (FRF), and $X(\omega)$ and $Y(\omega)$ are the Fourier transforms of $x(t)$ and $y(t)$. Assuming that both the input and the output are ergodic processes, applying (2.38), the PSD of the output is given by

$$S_{yy}(\omega) = \lim_{T\to\infty} \frac{2\pi}{T} \left|\hat{Y}_T(\omega)\right|^2 \tag{2.51}$$

where $\hat{Y}_T(\omega) = H(\omega)\hat{X}_T(\omega)$ idealizes the Fourier transform of a long sample of the output. Substituting (2.50) in this expression, it is obtained

$$S_{yy}(\omega) = |H(\omega)|^2 \lim_{T\to\infty} \frac{2\pi}{T} \left|\hat{X}_T(\omega)\right|^2 = |H(\omega)|^2 S_{xx}(\omega). \tag{2.52}$$

This important property of LTI systems indicates that the PSD of the output $Y(\omega) = H(\omega)X(\omega)$ of a dynamical system is simply obtained by multiplying the PSD of the input with the squared norm of the FRF. Equation (2.52) is is a general form valid for SDOF and extendable MDOF systems.

The derivation is very similar at third order, although based this time on Equation (2.46). It yields

$$B_{yyy}(\omega_1, \omega_2) = H(\omega_1) H(\omega_2) \overline{H}(\omega_1 + \omega_2) B_{xxx}(\omega_1, \omega_2). \tag{2.53}$$

showing that the bispectrum of the output is simply obtained by multiplication of the bispectrum of the input and a *kernel*, $H(\omega_1) H(\omega_2) \overline{H}(\omega_1 + \omega_2)$, sometimes called a Volterra kernel [23].

As a particular case, when the FRF is constant, e.g. $H = a$, the dynamical system is said to be *memoryless*. The impulse response function is a Dirac delta function, $h(t) = a\,\delta(t)$, and the system just corresponds to a static transformation as $y(t) = a\,x(t)$. In this case, the PSDs and bispectra of the input and output are simply related by

$$S_{yy}(\omega) = a^2 S_{xx}(\omega) \quad ; \quad B_{yyy}(\omega_1, \omega_2) = a^3 B_{xxx}(\omega_1, \omega_2) \tag{2.54}$$

which explicitly recalls that the PSD is a quadratic quantity and the bispectrum a cubic quantity.

2.3 Fundamentals of the Spectral and Bispectral Analyses

Section 2.2 reviewed the essential statistical concepts needed to address the introductory problem presented in Section 2.1. The goal of a structural analysis

is to determine the *structural responses*, which can include displacements, accelerations, internal forces, or ground reactions. Just as simulations of a random experiment can be used to determine the probability of rolling a "6" on a die, the statistics of structural responses can be studied using Monte Carlo simulations. This method is demonstrated first in Section 2.3.1, providing an intuitive understanding of the problem. However, it is less efficient compared to spectral and bispectral analyses, which are illustrated at the end of this section.

2.3.1 Monte Carlo solution of the problem

Monte Carlo simulations rely on numerical simulations of the problem, based on samples of the relevant stochastic processes. In this case, samples of the random fluctuations of the turbulent wind velocity $u(t)$ are generated to correspond with the known power spectral density (PSD) of the wind turbulence. The dynamical system can then be simulated using time-domain integrators [24], resulting in time series for the structural responses. These time series allow for the calculation of several statistical indicators.

For the numerical analysis, it is necessary to provide numerical values for all problem parameters. In this illustration, the parameters are chosen as follows: mass $m = 1000$kg, natural frequency $f_0 = 1.5$Hz, and damping ratio $\xi = 3\%$. The viscosity c and stiffness k can be determined using $\omega_0 = 2\pi f_0$, $k = m\omega_0^2 = 88.826 \cdot 10^3$N/m, and $c = 2m\omega_0\xi$. The drag coefficient c_D and exposed area Ω are chosen such that $\frac{1}{2}\rho c_D \Omega = 15$ kg/m. Lastly, the mean wind velocity is set to $U = 10$ m/s, and the PSD of the wind turbulence $u(t)$ follows the model proposed by Von Karman:

$$S_u(\omega) = \frac{L_u}{2\pi^2 U} \frac{\sigma_u^2}{\left(1 + \left(1.339 \frac{L_u \omega}{2\pi U}\right)^2\right)^{5/6}} \tag{2.55}$$

where $\sigma_u = 1.5$m/s and $L_u = 150$m represent the standard deviation and length scale of the wind turbulence respectively. This PSD is a function of the circular frequency ω and such that its integral over the entire real axis corresponds with the variance σ_u^2.

Based on these numerical values, samples of the wind turbulence can be generated using standard approaches [25]. The equation of motion is integrated with Newmark's unconditionally stable algorithm [14], where the time step is chosen as $\Delta t = 0.05s$. The number N of considered time steps varies from $2^{10} = 1024$ to $2^{20} \approx 10^6$.

Figure 2.4 shows an example of structural displacement. More precisely, the dynamic response (in blue) is the output of the dynamical system, while the quasi-static response (in red) consists of the generated loading $f(t)$ divided by

the structural stiffness. The dynamic response departs from the quasi-static one due to inertial and viscous effects. A structural design would likely focus on the maximum displacement, as this is the quantity of interest for an engineer. However, deriving intermediate statistics is important as it provides a much better understanding of the problem.

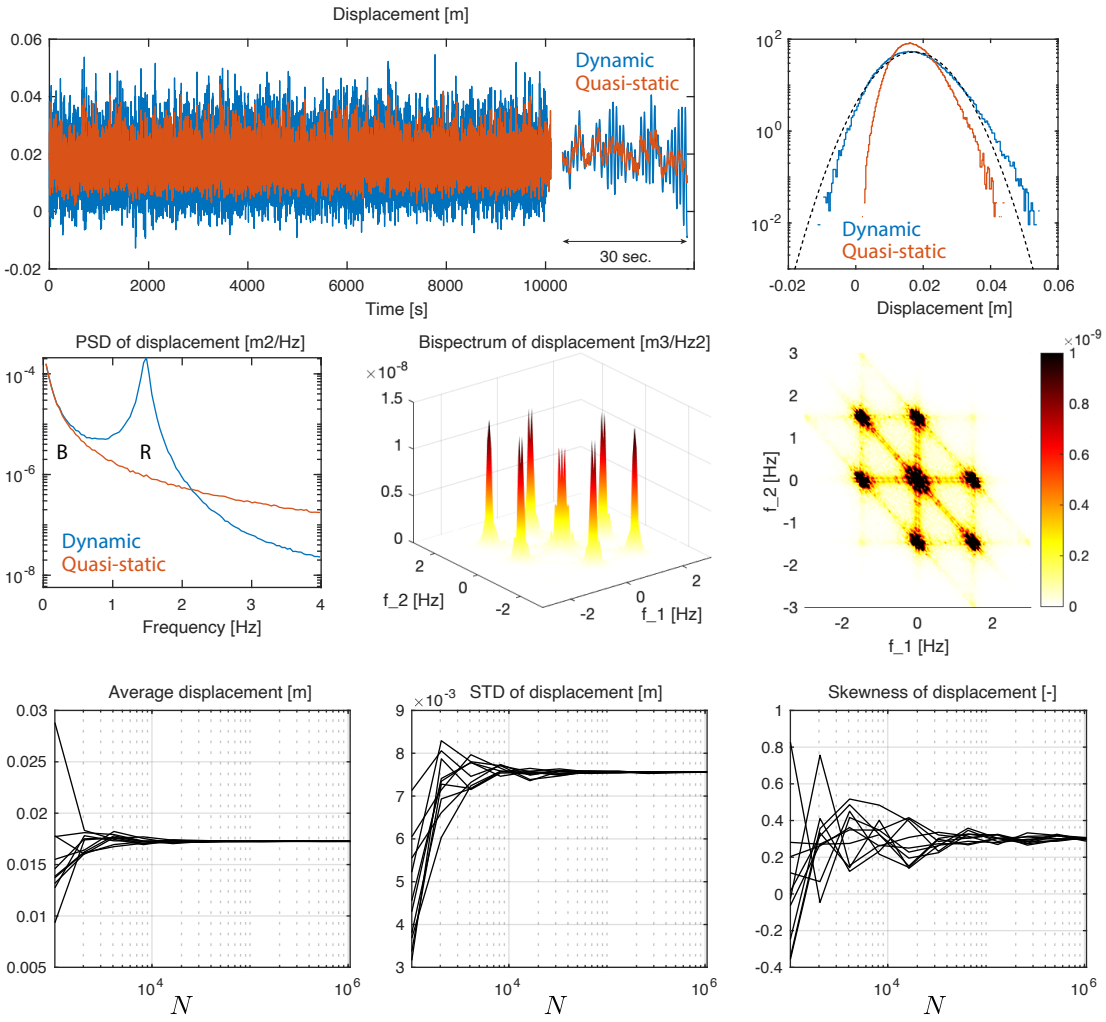


Figure 2.4: Solution of the introductory problem with a Monte Carlo approach : (top) time series of the displacement and its PDF, (middle) PSD of displacement and bispectrum, (bottom) sensitivity of estimated statistical indicators of the displacement.

Some additional basic statistics are provided in Figure 2.4. In the top right corner, the normalized histograms of the structural displacement under the two considered configurations (with and without dynamic effects) show that the structural response is non-Gaussian. The asymmetry in the dynamic response is less

pronounced than in the quasi-static response as a result of the central limit theorem. Indeed, the dynamic response at a given time is the convolution result of loads applied on the structure at several previous time instants, while the quasi-static (memoryless) response depends only on the load applied at the same time. Hence, the non-Gaussian nature of the quasi-static response is more pronounced.

The dashed curve represents the distribution of a hypothetical Gaussian response with the same mean and standard deviation as the dynamic response. It is observed that the actual PDF (in blue) extends more to the right than the Gaussian distribution. This will significantly impact the extreme values used for design.

The PSD of the response is another insightful quantity. It can be estimated with a spectrogram approach by implementing Equation (2.38) and averaging over several time windows. While the quasi-static response consists only of low-frequency components, resulting from the memoryless transformation of the slow turbulence, the dynamic response (in blue) also features a resonance peak near the natural frequency $f_0 = 1.5$ Hz. The labels "B" and "R" refer to the background (quasi-static response) and resonant (dynamic response) contributions to the total structural response.

Similarly, the bispectrum of the structural displacement can be computed by implementing Equation (2.46) with a spectrogram approach. This results in a background peak in the center of the frequency domain, plus six additional resonance peaks located around the central background peak. The PSD and bispectrum are insightful as they indicate that the response is approximately equally split between the background and resonant components. In other words, the dynamic amplification is roughly equal to two. However, this concept is not emphasized in this thesis, as the important design quantity is the extreme value (largest over a certain window), which results not only from an increase in standard deviation but also from the non-Gaussianity of the response.

The Monte Carlo analysis was repeated 10 times for various values of the number of time steps N. Results reported in the last line of Figure 2.4 illustrate the variability of the estimated statistics of the structural response. The average displacement is 0.0173 m, and Figure 2.4 shows that this value is accurately obtained with $N \gtrsim 10^4$. Using long time series, the standard deviation of the displacement is evaluated at 0.0076 m, requiring a signal with about $5 \cdot 10^4$ points to achieve significant accuracy. Lastly, the skewness coefficient of the displacement is 0.31, necessitating a very long time series to mitigate issues related to the larger standard error in the sampling distribution. The higher the order of the statistical indicator, the longer the required time series.

In this simple introductory problem, it is not difficult to simulate $N = 10^6$ time steps. However, in more realistic civil engineering problems, structures are modeled with thousands of degrees-of-freedom, responding in many structural

modes and spanning several timescales. This is why using Monte Carlo simulations to estimate higher-order statistics of structural responses is still not effective today. Just as it is not difficult to numerically roll a die many times to estimate the probability of getting a "6", but this is more expensive than simply dividing 1 by 6, the spectral and bispectral analyses avoid simulations and provide the statistics of the result through algebraic computations.

2.3.2 Mean response

The computation of the mean displacement is the first and simplest task that can be done in a probabilistic framework. It simply consists in averaging the equation of motion (2.1). Considering that the average acceleration and average velocity cannot be different from zero (otherwise this would result in a drift), averaging the equation of motion yields:

$$k\mathbb{E}\left[x(t)\right] = \mathbb{E}\left[f(t)\right] \qquad \rightarrow \qquad \mu_x = \frac{\mu_f}{k}, \tag{2.56}$$

so that the average displacement is just obtained as the ratio of the average loading μ_f and the structural stiffness k. It has been considered here that the loading is stationary so that the average loading is constant. It is obtained by averaging Equation (2.2):

$$\mu_f = \mathbb{E}\left[f(t)\right] = \frac{1}{2}\rho c_D \Omega \mathbb{E}\left[(U+u(t))^2\right] = \frac{1}{2}\rho c_D \Omega \left(U^2 + \sigma_u^2\right) = 1533.8\text{N}, \tag{2.57}$$

and the average displacement, obtained by dividing by the stiffness, gives for the chosen numerical values $\mu_x = 0.0173$m. This result is consistent with the result obtained with the Monte Carlo simulation. It is however obtained by means of a simple algebraic equation.

2.3.3 Second order

The same approach can be developed at second order. Since the problem involves a dynamical system, the filtering of the frequency content of the input plays a major role and the memoryless derivation used for the determination of the average requires adaptation.

Introducing the FRF of the problem $H(\omega) = (-m\omega^2 + i\omega c + k)^{-1}$, the PSD of the response is computed by application of Equation (2.52), and reads

$$S_x(\omega) = |H(\omega)|^2 S_f(\omega). \tag{2.58}$$

It is possible to prove that the PSD of the applied loading is given by [26]:

$$S_f(\omega) = \left(\frac{1}{2}\rho c_D \Omega\right)^2 \left[4U^2 S_u(\omega) + 2\int_{-\infty}^{+\infty} S_u(\omega_1)\, S_u(\omega - \omega_1)\, d\omega_1\right]. \tag{2.59}$$

The second term in this expression is one order of magnitude smaller than the first one and is usually neglected. Details on how to establish this expression will be given in Chapter 4 for more advanced configurations, particularly in a 3-D turbulence field and for realistic aerodynamic components.

Following (2.54), the expression $S_f(\omega)/k^2$ provides the PSD of the quasi-static component of the response $x(t)/k$. It accurately matches the red curve in the PSD plots of Figure 2.4, but is reported in Figure 2.5 in linear scales for better readability. The PSD of the dynamic response $S_x(\omega)$ is also reported in blue in Figure 2.5. The variance of the response is obtained by numerical integration of $S_x(\omega)$:

$$\sigma_x^2 = \int_{-\infty}^{+\infty} S_x(\omega)\, d\omega. \tag{2.60}$$

In a simple implementation, integration can be performed by spreading N_f integration points uniformly from -20 rad/s to 20 rad/s (values chosen to go far enough beyond the natural frequency of this particular example). This choice is far from optimal as extensively discussed in this thesis. However, the number $N_f \sim 100$ of integration points required to provide a converged estimate of the standard deviation of the response, 0.00755 m, is much lower than the number of (yet more costly) time steps required in the Monte Carlo approach. Again, this indicates that it is possible to provide a closed-form expression for the PSD of the response, which provides the standard deviation of the response after numerical integration.

One should not underestimate, though, the need to increase the number of integration points as damping decreases. Indeed, this numerical integration requires a large number of integration points in the neighborhood of the natural frequency, where large contributions to the variance are expected, and where, from standard structural dynamics, it is known that this peak is as sharp as the damping ratio is small.

Last but not least, it should be mentioned that the spectral analysis of SDOF systems has been applied since the mid-20th century. At times when computational power was less efficient than nowadays, integrating PSD with sharp peaks could be approached in an approximate analytical manner by taking advantage of the timescale separation and the sharpness of the peaks [27]. This has resulted in the well-known Background/Resonant decomposition [28], which is behind the

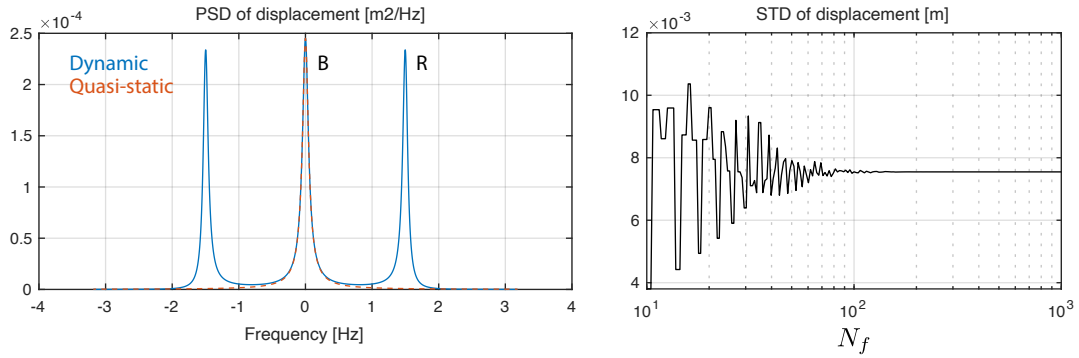


Figure 2.5: Second order solution of the introductory problem with a spectral approach : PSD of displacement and sensitivity of estimated statistical indicators with respect to number of integration points.

national codes and standards. Application of this theory to the problem with the notations at hand gives

$$\sigma_x^2 = \frac{1}{k^2}\left(\sigma_f^2 + S_u\left(\omega_0\right)\frac{\pi\omega_0}{2\xi}\right) \tag{2.61}$$

where $\sigma_f = 2U\left(\frac{1}{2}\rho c_D \Omega\right)\sigma_u$ is the standard deviation of the applied load. Application of this equation with the considered numerical values provides $\sigma_x = 0.00766$ m, which is slightly larger than the actual value, obtained with accurate numerical integration (0.00755 m).

2.3.4 Third order

It is already clear from the first two statistical orders that spectral analyses, being non-sampling-based methods, offer a fast way to compute the statistics of the response of a dynamical system subjected to a random input. This is reinforced at higher orders, since as shown before, longer Monte Carlo simulations are required to provide reliable estimates of higher order moments.

At third order, application of Equation (2.54) gives:

$$B_x\left(\omega_1,\omega_2\right) = H\left(\omega_1\right)H\left(\omega_2\right)\overline{H}\left(\omega_1+\omega_2\right)B_f\left(\omega_1,\omega_2\right) \tag{2.62}$$

where the bispectrum of the loading $B_f\left(\omega_1,\omega_2\right)$ is given by

$$\begin{aligned} B_f\left(\omega_1,\omega_2\right) = 8\left(\frac{1}{2}\rho c_D\Omega\right)^3 & \\ \left[S_u\left(\omega_1\right)S_u\left(\omega_2\right) + S_u\left(\omega_1+\omega_2\right)S_u\left(\omega_1\right) + \right. & \\ \left. + S_u\left(\omega_2\right)S_u\left(\omega_1+\omega_2\right)\right]. & \end{aligned} \tag{2.63}$$

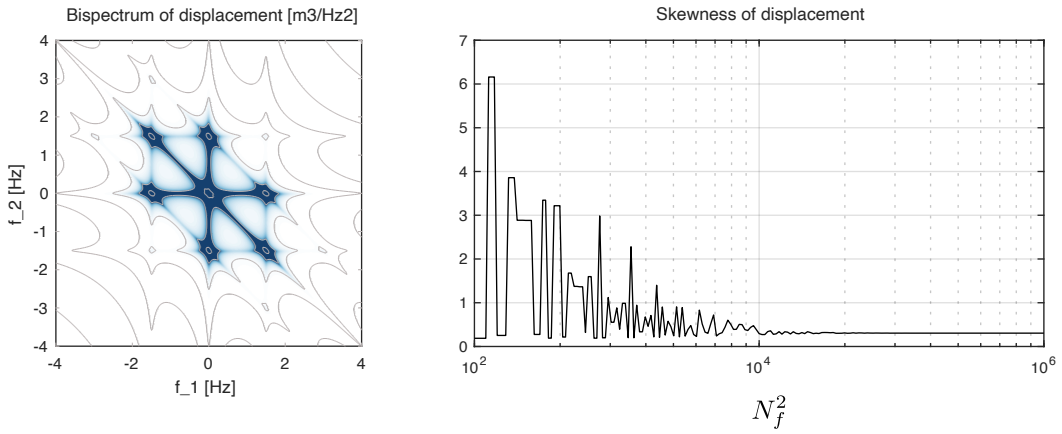


Figure 2.6: Third order solution of the introductory problem with a bispectral approach : bispectrum of displacement and sensitivity of estimated statistical indicator with respect to number of integration points.

This expression provides the leading order expression of the loading. Its establishment for this simple problem is available in the literature [29], and will be developed in Chapter 3 for more advanced configurations. The third moment of the response is obtained by numerical integration of $B_f(\omega_1, \omega_2)$:

$$m_{3,x} = \iint_{-\infty}^{+\infty} B_x(\omega_1, \omega_2)\, d\omega_1 d\omega_2. \tag{2.64}$$

The first difficulty consists in integrating on a 2-D frequency space, which hints at issues related to the curse of dimensionality. The background peak, located in the neighborhood of the origin, as well as the six resonance peaks, need a fine mesh to be accurately integrated. At this stage, a regular mesh is used for illustration. A large number of sampling points can be used for an accurate estimation of this integral, but also of the bispectrum of the response. With $N_f = 10^3$ in each direction of the frequency space, a very accurate representation of the bispectrum of the response can be obtained. It is shown in Figure 2.6 as contour lines. The greyed contours in that figure correspond to equally spaced values on a log scale (i.e., down to very small values) and the six-arm star shown in the middle indicates where the bispectrum is large, and the most important contributions are expected.

Using N_f sampling points for both independent variables, ω_1 and ω_2, results in a meshing with N_f^2 integration points in total. The result of the numerical integration, $m_{3,x}$, is shown as a function of N_f^2 in Figure 2.6. Again, about 100 sampling points in each direction seem to provide an accurate estimate of the skewness of the response, with a limiting value of $\gamma_{3,x} = 0.305$. This, again, agrees very well with the Monte Carlo simulation, and very significant discrepancies can

be obtained if the meshing is too scarce, up to a factor 20 (!). Also, it is noticed that the skewness of the response is directly obtained from the time series in the Monte Carlo approach, while it is obtained as the integral of the bispectrum in the bispectral approach.

2.3.5 Extreme response

The end scope of a structural analysis, in view of an engineering design, is to provide the maximum values of the structural responses. These can indeed be compared to acceptable limits to verify a design. Therefore, the final question of the structural analysis consists in the determination of the statistics of the extreme values of stochastic processes, that is, for a conventionally given period of time (e.g., 600 seconds for wind generation), to determine the PDF of the largest value which could be observed in the process.

This question is easily dealt with in a Monte Carlo simulation process, since it requires the repeated simulation of samples of the responses. For each sample, the maximum response can be identified in the time series, and the PDF of the extreme value simply consists in the distribution of this collection of values.

In a spectral approach, several theories have been developed to approach the PDF of extreme values of stochastic processes over a given time window [13]. The detailed establishment of these theories goes beyond the scope of this work, but from a very practical point of view, they aim at estimating the maximum response with a formulation such as:

$$x_{\max} = \mu_x + g_x \sigma_x \tag{2.65}$$

where g_x is the so-called *peak factor*. It is a number, usually larger than 3 [30], indicating how many standard deviations need to be added to the average response in order to create a design quantity. The quantity $x_{\max}$ in fact represents the average of the distribution of the extreme value of the process encountered on a time window T. A commonly accepted model for Gaussian processes [31] gives

$$g_x = \beta + \frac{\gamma}{\beta} \tag{2.66}$$

where $\gamma \approx 0.5772$ is Euler's constant, and $\beta = \sqrt{2 \ln \nu_0 T}$, where ν_0 is the so-called *zero-upcrossing frequency*, i.e., the average frequency at which the random process passes through its mean value. This frequency can be obtained from the spectral moments of the PSD [13],

$$\nu_0 = \frac{1}{2\pi} \left(\frac{\int_{-\infty}^{+\infty} \omega^2 S_u(\omega)\, d\omega}{\int_{-\infty}^{+\infty} S_u(\omega)\, d\omega} \right)^{1/2}. \tag{2.67}$$

Theories leading to the establishment of this simple formula are quite complicated, even for Gaussian processes. The current state-of-the-art for non-Gaussian processes is to model them as cubic transformations of some underlying Gaussian processes. The idea is to determine the coefficients a_i, $i = 0, \ldots, 3$, of a cubic (memoryless) transformation,

$$x(t) = a_3 u^3(t) + a_2 u^2(t) + a_1 u(t) + a_0 \tag{2.68}$$

in such a way that this transformation of the normalized centered Gaussian process $u(t)$ provides the actual statistical properties of the Gaussian process $x(t)$. Since this transformation is memoryless and restrained to be monotonic, there is a one-to-one correspondence between the extreme values of $u(t)$ and those of $x(t)$. As a consequence, the distribution of the extremes of the non-Gaussian process $x(t)$ can be expressed as a function of those of $u(t)$. After detailed mathematical derivations [30], another simple formulation is finally obtained for g_x:

$$\begin{aligned} g_{\mathrm{NG}} = k \Bigg\{ & \left(\beta + \frac{\gamma}{\beta}\right) + h_3 \left[\beta^2 + (2\gamma - 1) + \frac{1.98}{\beta^2}\right] \\ & + h_4 \left[\beta^3 + 3\beta(\gamma - 1) + \frac{3}{\beta}\left(\frac{\pi^2}{6} - \gamma + \gamma^2\right) + \frac{5.44}{\beta^3}\right] \Bigg\} \end{aligned} \tag{2.69}$$

where

$$h_3 = \frac{\gamma_3}{4 + 2\sqrt{1 + 1.5\gamma_e}}, \quad h_4 = \frac{\sqrt{1 + 1.5\gamma_e} - 1}{18}, \tag{2.70}$$

γ_3 is the skewness coefficient of the process, and $\gamma_e = \gamma_4 - 3$ is the excess coefficient (4th order), and where $k = 1/\sqrt{1 + 2h_3^2 + 6h_4^2}$. This formulation degenerates into the Gaussian formulation of Equation (2.66) when $\gamma_3 = \gamma_e = 0$, in the case of a Gaussian process. Since this work focuses on the determination of the skewness coefficient γ_3, linked to the second and third order statistical moments, there is currently no access to the fourth moment, and to γ_e, which prevents the formal application of this theory. However, Rigo et al. [32] and Denoël et al. [33] have observed that the non-Gaussian processes encountered in wind engineering applications where polynomial transformation of the wind turbulence velocity is used, lie in the near vicinity of the so-called monotonic limit

$$\gamma_e \simeq (1.25\gamma_3)^2 \tag{2.71}$$

limiting, in the (γ_3, γ_e)-plane, the values corresponding to a monotonic cubic transformation (allowing the one-to-one correspondence). Therefore, lacking information about γ_e, it will be assumed in the following that the problem sits on this limit, and that the expression of γ_e is used in (2.69) to determine the peak factor.

Last but not least, non-Gaussian processes are characterized by (possibly) asymmetric distributions. This hints that maximum and minimum extreme values are not located symmetrically around the mean value as soon as the skewness coefficient $\gamma_3 \neq 0$. The formulation proposed in [30] therefore develops in two versions,

$$x_{\mathrm{NG}_+} = \overline{x} + g_{\mathrm{NG}_+}\sigma_x; \quad x_{\mathrm{NG}_-} = \overline{x} - g_{\mathrm{NG}_-}\sigma_x \tag{2.72}$$

where the two non-Gaussian peak factors are obtained by considering the two possible signs for the skewness coefficient in Equation (2.69).

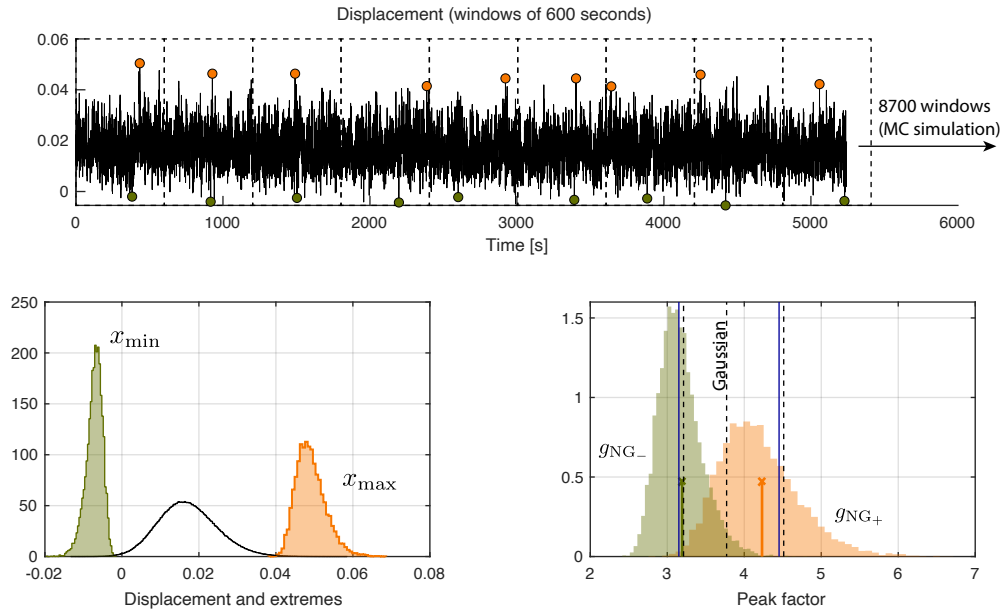


Figure 2.7: Illustration of the determination of extreme values (of non-Gaussian processes). (Top) a time series segmented in several windows. (Bottom) distribution of maximum and minimum values, and of peak factors.

Figure 2.7 illustrates these concepts with the considered problem. On the top, a representative time series is shown. It was simulated long enough so that 8700 windows of 600 seconds each could provide one sample of the maximum and minimum values of the displacement over each window. This collection of 8700 samples required the simulation of more than 10^8 time steps, but offered the informative distributions of the extreme values shown for both the minimum and maximum values. To guide the eye, the distribution of the displacement is also represented (in black). This information was possible to obtain because the considered problem is just an SDOF problem; this detailed information is typically not affordable in a realistic application, by means of a Monte Carlo approach.

The bottom right plot shows the distributions of extreme values, but expressed in terms of peak factors, positive and negative. The dashed line in the middle provides the unique peak factor (same for min and max values) that would be obtained in a Gaussian framework. Applying Equation (2.66), it has been found $g_x = 3.77$, which significantly differs from the mean peak factors obtained with the long Monte Carlo simulations, namely 3.19 and 4.23, respectively, for both negative and positive sides. These two values are represented by the thick colored crosses located at the centroids of the distributions. Application of the cubic translation model provides 3.21 and 4.51 (shown with dashed lines), while application of the simplified proposition based on (2.71) for the estimation of γ_e yields respectively 3.15 and 4.45 (shown with solid lines) for the two peak factors.

This illustration highlights several facts:

- The minimum and maximum responses observed over a given period of time are random variables.
- So are the corresponding peak factors.
- The actual distribution of the peak factor is difficult to estimate for larger-sized problems which cannot afford very long simulations.
- Non-Gaussian processes have different positive and negative peak factors, while the Gaussian assumption provides an intermediate estimate.
- The cubic translation model is not perfect but performs better than the Gaussian assumption (4.45 instead of 3.77 for g_x).
- γ_3 of about 0.3, yields a discrepancy of about 18% on the peak factor that would be obtained with a Gaussian analysis.

2.4 Scalability

When the general equation of motion of a dynamic system is tackled in a stochastic manner in the frequency domain, the analysis becomes significantly more complex if the system responses are non-Gaussian due to non-Gaussian loading. This complexity arises because each response (such as displacements, internal forces, ground reactions, etc.) will have corresponding bispectra.

Each of these bispectra might exhibit very sharp peaks, making their accurate capture challenging. Accurate capture is crucial because the bispectra must be integrated over the frequency space to estimate the third statistical moments, which in turn will provide information on the statistical asymmetry, and affect

extreme values for design. Poor resolution of these peaks leads to inaccurate integrals and, consequently, incorrect statistical moments, which result in erroneous predictions of the system's behavior.

The issue of scalability becomes particularly pressing as the system's complexity increases. When dealing with systems that have thousands of degrees of freedom (DOFs), the computational effort required to establish and analyse the high-order spectra, such as bispectra, becomes extremely time-consuming. This is a significant challenge because the number of response bispectra increases with the number of DOFs, and each one needs to be accurately resolved and integrated.

At the inception of this research, the largest problem that had been solved involved only a system with 7 degrees of freedom [12]. Extending these methods to systems with thousands of degrees of freedom presents a formidable challenge due to the more-than-proportional increase in computational burden. The sheer volume of data and the necessity for high-resolution spectral capture demand substantial computational resources and sophisticated algorithms to manage the workload efficiently.

To address these scalability issues, this thesis has contributed to the development of more efficient computational methods and algorithms. These methods aimed at reducing the time and computational power required for bispectral analysis while maintaining the precision necessary for reliable statistical moment estimation. Developing efficient methods to tackle these challenges was crucial for advancing the field and enabling accurate analysis of more complex structures.

Chapter 3

Bispectral Analysis of multi degree-of-freedom structures

3.1 The general equations of motion of a multi-degree-of-freedom system

Chapter 2 tackled the Bispectral problem of a single-degree-of-freedom (SDOF) system. Though one might argue that real civil structures have far more than a single degree-of-freedom, it served to introduce the Bispectral problem and the most important concepts related to it in the most simple but concrete possible way.

This Chapter will extend the concept of Chapter 2 to multi degree-of-freedom (MDOF) structures.

The equation of motion of a MDOF dynamic system reads

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{f}(t) \tag{3.1}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are the `NDOFs` $\times$ `NDOFs` structural matrices, $\mathbf{f}(t)$ the `NDOFs` $\times$ `1` vector of applied forces (wind forces) at time instant t, $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ the `NDOFs` $\times$ `1` vectors of structural displacements, velocities and accelerations respectively. `NDOFs` represents the total number of structural degrees-of-freedom.

Equation (3.1) and all following ones will assume a Finite Element Method approach to the problem, where notably the continuous system is discretised using nodes and elements [34, 35, 36].

In Civil Engineering, it is common practice to project the equations of motion in (3.1) onto a special, orthogonal basis, called *modal basis*, intrinsic to the structural system. As such, the modal equations of motion read

$$\mathbf{M}^{\star}\ddot{\mathbf{q}}(t) + \mathbf{C}^{\star}\dot{\mathbf{q}}(t) + \mathbf{K}^{\star}\mathbf{q}(t) = \mathbf{p}(t) \tag{3.2}$$

where $\mathbf{M}^\star$, $\mathbf{C}^\star$ and $\mathbf{K}^\star$ are the $\texttt{NM}\times\texttt{NM}$ modal mass, damping and stiffness matrices

$$\mathbf{M}^\star = \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}; \quad \mathbf{C}^\star = \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi}; \quad \mathbf{K}^\star = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} \tag{3.3}$$

$\mathbf{\Phi}$ the $\texttt{NDOFs} \times \texttt{NM}$ structural modal matrix, i.e. the ensemble of the eigenvectors of the orthogonal basis, $\mathbf{p}(t) = \mathbf{\Phi}^T\mathbf{f}(t)$ the modal loads and $\mathbf{q}(t)$ the modal responses. NM represents the total number of structural modes (i.e. eigenvectors) kept in the definition of this reduced modal basis. Generally $\texttt{NM} \ll \texttt{NDOFs}$, so that the determination of such basis is convenient and the projection of Equation (3.1) to obtain Equation (3.2) is accompanied by a substantial reduction of the size of the subsequent problem. This is specifically true for very large structures, for which the total number of degrees-of-freedom NDOFs can exceed tens of thousands, while the usual number of kept vibration modes NM would limit to a maximum of about one hundred. Moreover, another possible advantage of such projection stands in the fact that, if the system is assumed with proportional damping (e.g. the damping matrix is obtained as a linear combination of the mass and stiffness matrices, also known as Rayleigh damping [14]),

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

so that not only $\mathbf{M}^\star$ and $\mathbf{K}^\star$ are diagonal (from the way $\mathbf{\Phi}$ is defined), but also $\mathbf{C}^\star$. Equation (3.2) results into a set of NM independent differential equations. This configuration is indeed much more optimal than having to solve a coupled system of equations, since doing so numerically would involve matrix inversion operations which are more expensive, specially when it comes to large MDOF problems. Nonetheless, application of Equation (3.2) is valid in more general cases, including those where non-proportional damping models are employed in the structural system and the modal matrix $\mathbf{\Phi}$ is obtained through the solution of the generalised eigenvalue problem [37, 38].

In such context, solving Equation (3.2) can be generally done in (i) time domain by means of classical time-stepping algorithms [24], or (ii) frequency domain by means of Fourier Transforms, in which case Equation (3.2) reads

$$\mathbf{Q}(\omega) = \mathbf{H}^\star(\omega)\mathbf{P}(\omega) \tag{3.4}$$

where $\mathbf{H}^\star(\omega)$ is the complex modal Frequency Response Function (FRF)

$$\mathbf{H}^\star(\omega) = \left(-\mathbf{M}^\star\omega^2 + i\omega\mathbf{C}^\star + \mathbf{K}^\star\right)^{-1}, \tag{3.5}$$

$\mathbf{P}(\omega)$ and $\mathbf{Q}(\omega)$ being the Fourier Transforms of modal loads and responses respectively. In the nodal basis, the Frequency Domain equations of motion read

$$\mathbf{X}(\omega) = \mathbf{H}(\omega)\mathbf{F}(\omega) \tag{3.6}$$

where

$$\mathbf{H}(\omega) = \left(-\mathbf{M}\omega^2 + i\omega\mathbf{C} + \mathbf{K}\right)^{-1} \tag{3.7}$$

is the complex nodal Frequency Response Function.

3.2 Statistics and Probability

Section 2.2 covered the most important concepts of statistics and probability of single random variables and processes. This Section will extend those concepts to the cases of multiple random variables and processes.

3.2.1 Multiple random variables

3.2.1.1 Joint Probability Density Function

Let's first consider two random variables X_1 and X_2. Each one of them can be marginally characterised with the concepts discussed in Section 2.2.1. However, doing so would not reveal any possible interrelationship between the two random variables. This additional information is given by the *joint* distribution of X_1 and X_2. The joint cumulative density function is defined as [13]

$$F_{X_1X_2}(x_1, x_2) = P\left[\{X_1 \leq x_1\} \cap \{X_2 \leq x_2\}\right]. \tag{3.8}$$

Similarly, the joint probability density function is defined as

$$p_{X_1X_2}(x_1, x_2) = \frac{\partial^2}{\partial x_1 \partial x_2} F_{X_1X_2}(x_1, x_2) \tag{3.9}$$

and represents the probability that the two random variables X_1 and X_2 are simultaneously in the ranges $(x_1; x_1 + dx_1]$ and $(x_2; x_2 + dx_2]$ respectively. The reciprocal relationship is

$$F_{X_1X_2}(x_1, x_2) = \int_{-\infty}^{x_1}\int_{-\infty}^{x_2} p_{X_1X_2}(y_1, y_2)dy_1dy_2. \tag{3.10}$$

These concepts are easily extended to n random variables:

$$F_{X_1\ldots X_n}(x_1, \ldots, x_n) = P\left[\{X_1 \leq x_1\} \cap \cdots \cap \{X_n \leq x_n\}\right], \tag{3.11}$$

$$p_{X_1\ldots X_n}(x_1, \ldots, x_n) = \frac{\partial^n}{\partial x_1 \ldots \partial x_n} F_{X_1\ldots X_n}(x_1, \ldots, x_n). \tag{3.12}$$

3.2.1.2 Joint moments

In Section 2.2.1, Equation (2.6) provided the general definition of the $n-$th order statistical moment of a random variable. When multiple random variables are considered, one defines the so called *joint moments*. Considering the simple example of two random variables X_1 and X_2, the joint moment of order $n+m$ is defined as [13]

$$\mathbb{E}[X_1^n X_2^m] = \int_{-\infty}^{\infty} x_1^n x_2^m p_{X_1X_2}(y_1, y_2) dy_1 dy_2. \tag{3.13}$$

Similarly, the *joint central moment* of order $n+m$ is defined as

$$\mathbb{E}[(X_1 - \mu_{X_1})^n (X_2 - \mu_{X_2})^m] = \int_{-\infty}^{\infty} (x_1 - \mu_{X_1})^n (x_2 - \mu_{X_2})^m p_{X_1X_2}(y_1, y_2) dy_1 dy_2. \tag{3.14}$$

Specialising Equation (3.14) for $n = m = 1$, is what defines the *covariance* of the two random variables X_1 and X_2:

$$\sigma^2_{X_1X_2} = \mathbb{E}[(X_1 - \mu_{X_1})(X_2 - \mu_{X_2})] = \mathbb{E}[X_1X_2] - \mu_{X_1}\mu_{X_2} \tag{3.15}$$

Equations (3.13) and (3.14) can be easily extended to the general case of n random variables. For instance, later in this Work, third order cross moments such as $\mathbb{E}[X_1X_2X_3]$ or $\mathbb{E}[X_1X_2^2]$ will be treated.

3.2.2 Multiple random processes

In this Section, the concepts introduced in Section 2.2.2 will be extended to the case of multiple random processes. To give a concrete example of multiple random processes, one might think of the measurements of the wind speed at different points in space on a long-span bridge.

To be concise, only stationary, ergodic random processes will be treated, so that focus is given to the mathematical concepts that will be extensively used throughout this Thesis. As a consequence, any ensemble mean will be automatically replaced by sample mean of any arbitrary realisation of the random processes, as discussed in Section 2.2.5.

3.2.2.1 The Cross-Correlation function

The Cross-Correlation function of two stationary, ergodic, random processes $X(t)$ and $Y(t)$ is defined as

$$R_{xy}(\tau) = \mathbb{E}[(X(t)-\mu_X)(Y(t+\tau)-\mu_Y)] = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} (x(t)-\mu_X)(y(t+\tau)-\mu_Y) dt. \tag{3.16}$$

Like the Autocorrelation function (see Equation (2.29)), if evaluated at zero lag $\tau = 0$, the Cross-Correlation function returns the *covariance* of the two random processes (normalised to their mean values):

$$\begin{aligned} \tilde{m}_{2,XY} = \sigma^2_{XY} &= \mathbb{E}[(X(t) - \mu_X)(Y(t) - \mu_Y)] \\ &= \int_{-\infty}^{\infty} (x(t) - \mu_X)(y(t) - \mu_Y) dt \\ &= m_{2,XY} - \mu_X \mu_Y \end{aligned} \tag{3.17}$$

where $m_{2,XY} = \mathbb{E}[X(t)Y(t)]$ is the second order cross moment.

For zero-mean processes, the relation $\tilde{m}_{2,XY} \equiv m_{2,XY}$ applies. To simplify notations even further, in the following only zero-mean (yet stationary and ergodic) random processes will be assumed, assuming the equivalence of normal and centred statistical moments of any order.

3.2.2.2 The Cross Power Spectral Density function

The Cross Power Spectral Density function (CPSD) is defined as the Fourier Transform of the Cross-Correlation function:

$$\begin{aligned} S_{xy}(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) y(t+\tau) dt \right] e^{-i\omega\tau} d\tau \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} R_{xy}(\tau) e^{-i\omega\tau} d\tau. \end{aligned} \tag{3.18}$$

The Cross-Correlation and the CPSD also form a Fourier Pair, so that, the integration of the CPSD gives the covariance of the two random processes:

$$\sigma^2_{XY} = \int_{-\infty}^{\infty} S_{xy}(\omega) d\omega. \tag{3.19}$$

3.2.2.3 The Cross-Bicorrelation function

If non-Gaussian random processes are considered, then mean and Autocorrelation function are not sufficient anymore for a complete characterisation of the processes.

Limiting discussions to third statistical order, the Cross-Bicorrelation function of three stationary, ergodic, random processes $X(t)$, $Y(t)$ and $Z(t)$ is defined as:

$$R_{XYZ}(\tau_1, \tau_2) = \mathbb{E}[X(t)Y(t+\tau_1)Z(t+\tau_2)] = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) y(t+\tau_1) z(t+\tau_2) dt. \tag{3.20}$$

It translates the existing triple correlation between, for instance, wind pressures at three different points in space, with time delays τ_1 and τ_2 from a reference time instant t.

Evaluating the Cross-Bicorrelation at both zero time lags gives the third-order cross moment

$$m_{3,XYZ} = R_{XYZ}(0,0) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)y(t)z(t)dt. \tag{3.21}$$

3.2.2.4 The Cross-Bispectrum

The Cross-Bispectrum is derived taking the twofold Fourier Transform of the Cross-Bicorrelation function:

$$B_{XYZ}(\omega_1,\omega_2) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} R_{XYZ}(\tau_1,\tau_2) e^{-i\omega_1\tau_1} e^{-i\omega_2\tau_2} d\tau_1 d\tau_2. \tag{3.22}$$

It should be now easy to understand that Cross-Bicorrelation and Cross-Bispectrum form a double Fourier Pair, and that integrating the Cross-Bispectrum in the 2D frequency space estimates the third order cross moment:

$$m_{XYZ} = R_{XYZ}(0,0) = \iint_{-\infty}^{\infty} B_{XYZ}(\omega_1,\omega_2) d\omega_1 d\omega_2. \tag{3.23}$$

3.2.3 System I/O relations

In this Section, the concepts introduced in Section 2.2.8 will be extended to MDOFs systems. They constitute the essential steps to do simple linear mathematical operations on vector processes. Let's assume a deterministic causal Linear Time Invariant (LTI) dynamical system defined by its Frequecny Response Function $\mathbf{H}(\omega)$ so that vectorial input $\mathbf{x}(t)$ and a vectorial output $\mathbf{y}(t)$ are related by

$$\mathbf{Y}(\omega) = \mathbf{H}(\omega)\,\mathbf{X}(\omega) \tag{3.24}$$

where $\mathbf{X}(\omega)$ and $\mathbf{Y}(\omega)$ are respectively the Fourier transforms of $\mathbf{x}(t)$ and $\mathbf{y}(t)$. Equation (3.24) can be specialised to any two components of $\mathbf{Y}(\omega)$, for instance $Y_m(\omega)$ and $Y_n(\omega)$. Extending the definition of the Cross Power Spectral Density function (ese Eq. (3.18)) to ergodic processes, as it was done in Section 2.2.8, we have

$$S_{y_{mn}}(\omega) = \lim_{T\to\infty} \frac{2\pi}{T} \hat{Y}_m(\omega)\,\overline{\hat{Y}_n}(\omega) \tag{3.25}$$

where $\hat{Y}_m(\omega)$ and $\hat{Y}_n(\omega)$ are the Fourier transforms of two idealized very long samples of $y_m(t)$ and $y_n(t)$ defined on the window $t \in [0;T]$. Substituting

$$\hat{Y}_m(\omega) = \sum_i H_{mi}(\omega)\,\hat{X}_i(\omega) \quad ; \quad \hat{Y}_n(\omega) = \sum_j H_{nj}(\omega)\,\hat{X}_j(\omega) \tag{3.26}$$

in (3.25), where the sums extend to the number of input in $\mathbf{x}(t)$, we have

$$\begin{aligned} S_{y_{mn}}(\omega) &= \sum_i \sum_j H_{mi}(\omega)\,\overline{H_{nj}}(\omega) \lim_{T\to\infty} \frac{2\pi}{T} \hat{X}_i(\omega)\,\overline{\hat{X}_j}(\omega) \\ &= \sum_i \sum_j H_{mi}(\omega)\,\overline{H_{nj}}(\omega)\,S_{x_{ij}}(\omega)\,. \end{aligned} \tag{3.27}$$

At second order, this expression can be written in the matrix format

$$\mathbf{S_y}(\omega) = \mathbf{H}(\omega)\,\mathbf{S_x}(\omega)\,\overline{\mathbf{H}}(\omega)\,, \tag{3.28}$$

which indicates that the CPSD matrix of the output is just expressed by left- and right-multiplication of the CPSD matrix of the input by the FRF. This expression extends to the MDOF case the relation (2.50) which had been obtained in the scalar case. In case of a memoryless transformation, $\mathbf{H}(\omega) \equiv \mathbf{A} = \text{cst}$, $\mathbf{y}(t) = \mathbf{A}\mathbf{x}(t)$, and

$$\mathbf{S_y}(\omega) = \mathbf{A}\mathbf{S_x}(\omega)\,\overline{\mathbf{A}}. \tag{3.29}$$

This operation is central in the projection or combination operations that are repeatedly used in the sequel.

At third order, following similar derivations, the cross-bispectrum of a triplet of components (m,n,o) of the output $\mathbf{y}(t)$ reads

$$\begin{aligned} B_{y_{mno}}(\omega_1,\omega_2) &= \sum_i \sum_j \sum_k H_{mi}(\omega_1)\,H_{nj}(\omega_2)\,\overline{H_{ok}}(\omega_1+\omega_2) \\ &\qquad \lim_{T\to\infty} \frac{2\pi}{T} \hat{X}_i(\omega_1)\,\hat{X}_j(\omega_2)\,\overline{\hat{X}_k}(\omega_1+\omega_2) \\ &= \sum_i \sum_j \sum_k H_{mi}(\omega_1)\,H_{nj}(\omega_2)\,\overline{H_{ok}}(\omega_1+\omega_2)\,B_{x_{ijk}}(\omega_1,\omega_2)\,. \end{aligned} \tag{3.30}$$

This relation will prove helpful when establishing the cross-bispectra of modal responses in different modes, or could be useful in other contexts where analysis is performed in a nodal basis. This expression shows that the establishment of a single triplet (m,n,o) of the response requires the combination of all cross-bispectra of the input in all possible combinations. Repeating this operation for all triplets and for many couples (ω_1,ω_2) shows why the analysis of large structural systems is computationally challenging.

In case of a memoryless system, $\mathbf{H}(\omega) \equiv \mathbf{A} = \text{cst}$, the former becomes

$$B_{y_{mno}}(\omega_1, \omega_2) = \sum_i \sum_j \sum_k A_{mi} A_{nj} \overline{A_{ok}} B_{x_{ijk}}(\omega_1, \omega_2) . \tag{3.31}$$

None of the two latter equations can be put in a simple matrix form. A tensor notation would be required. In the rest of this document, a component-wise notation similar to (3.31) will be used. Equations (3.29) and (3.31) extend to the MDOF case the same equations derived in Section 2.2.8 in the SDOF case.

3.3 Stochastic dynamic analysis

The general equations of motion (3.1), or (3.2), can be solved in either time or frequency domain, and each approach has its advantages and disadvantages. For the time domain, if its range of applicability includes also nonlinear problems (in both the loading process and/or the structural behaviour), its employment is only possible when records, measurements or simulations of loading time series are available. This is in some cases a limitation, either because of lack of measurement instrumentation or because of limited computing power, specially for the most complex (and scientifically interesting) applications. Nonetheless, if advances in technology and computing power are making the latter issue almost completely solved nowadays, the former is still a challenge in most cases, also considering the costs that modern innovative monitoring instrumentation require, for both installation and maintenance. Moreover, as shown in Section 2.3.1 (see Figure 2.4), a Monte Carlo approach to the problem carries an intrinsically high variability of accuracy of results, due to its intrinsic deterministic nature. To avoid such issues and increase the confidence interval, an optimal simulation/measuring time must be chosen, which might in some cases be considerably long.

On the other hand, a frequency domain approach to the resolution of the equations of motion is a serious alternative to these conventional time domain solutions, in that it allows the quantification of the global structural response. Besides, an advantage of such an approach stands in the fact that, if it can be certainly seen as a deterministic approach (i.e. by direct application of the Fourier Transform of a time series realisation), it has an (intrinsic) probabilistic (i.e. stochastic) version as well.

In a stochastic dynamic analysis the equations of motion are solved in a probabilistic sense. Knowing the probabilistic description of the loading, the aim is to determine the one of the structural response, that is, solving Equation (3.1) where the loading is known by means of probabilistic quantities, i.e. its CPSD

and Cross-Bispectra matrices, and the objective is to compute the same quantities for the structural responses.

Let's momentarily assume that the average wind loading $\overline{\mathbf{f}}$ is identified, and that both the CPSD and Cross-Bispectra matrices of the wind loading fluctuations around the mean are known.

3.3.1 First order

If the mean component of the wind load is identified, it can be taken out from the general equation of motion (see Equation (3.1)), so that

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\left(\overline{\mathbf{x}} + \tilde{\mathbf{x}}(t)\right) = \overline{\mathbf{f}} + \tilde{\mathbf{f}}(t) \tag{3.32}$$

where both the wind forces $\mathbf{f}(t)$ and the structural responses $\mathbf{x}(t)$ vectors have been split into their mean component $\overline{[\cdot]}$ and zero-mean fluctuating component $\tilde{[\cdot]}$.

Applying the expectation operator on both sides:

$$\mathbb{E}\left[\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t)\right] = \mathbb{E}\left[\mathbf{f}(t)\right] \tag{3.33}$$

the governing equations of motion reduce to the static case:

$$\mathbf{K}\overline{\mathbf{x}} = \overline{\mathbf{f}} \tag{3.34}$$

or

$$\mathbf{K}^{\star}\overline{\mathbf{q}} = \overline{\mathbf{p}} \tag{3.35}$$

in the modal basis, so that

$$\mu_{\mathbf{x}} \coloneqq \overline{\mathbf{x}} = \mathbf{K}^{-1}\overline{\mathbf{f}} \tag{3.36}$$

$$\mu_{\mathbf{q}} \coloneqq \overline{\mathbf{q}} = \mathbf{K}^{\star -1}\overline{\mathbf{p}} \tag{3.37}$$

link the average structural responses average applied loads.

3.3.2 Second order

In Section 3.1, Equation (3.4) introduced the frequency domain dynamic modal equation of motion of a MDOF system.

Considering Equation (3.4), and considering the mathematical relation between the Fourier Transform and the PSD of a random process

$$S_x(\omega) = \mathbb{E}\left[X(\omega)\overline{X}(\omega)\right] \tag{3.38}$$

the CPSD matrix of modal responses reads

$$\begin{aligned} S_{\mathbf{Q}}(\omega) &= \mathbb{E}\left[\mathbf{H}^{\star}(\omega)\mathbf{P}(\omega)\overline{\mathbf{H}^{\star}(\omega)\mathbf{P}(\omega)}\right] \\ &= \mathbb{E}\left[\mathbf{H}^{\star}(\omega)\mathbf{P}(\omega)\overline{\mathbf{P}}(\omega)\overline{\mathbf{H}^{\star}}(\omega)\right] \\ &= \mathbf{H}^{\star}(\omega)\mathbb{E}\left[\mathbf{P}(\omega)\overline{\mathbf{P}}(\omega)\right]\overline{\mathbf{H}^{\star}}(\omega) \\ &= \mathbf{H}^{\star}(\omega)\mathbf{S}_{\mathbf{P}}(\omega)\overline{\mathbf{H}^{\star}}(\omega) \end{aligned} \tag{3.39}$$

Equation (3.39) represents the second order Stochastic dynamic equations of motion, for multi-degrees-of-freedom (MDOFs) systems.

Then, following Equation (2.37), integration along the 1D frequency space of Equation (3.39) gives the second order statistical moments (i.e. variances and covariances) of the resulting stochastic modal responses $\mathbf{S}_{\mathbf{Q}}(\omega)$

$$\mathbf{m}_{2_{\mathbf{Q}}} = \Sigma_{\mathbf{Q}} = \int_{-\infty}^{\infty} \mathbf{S}_{\mathbf{Q}}(\omega)d\omega \tag{3.40}$$

Finally, statistics of structural responses are obtained recombining the modal responses obtained with application of Equation (3.39), that is

$$\mathbf{m}_{2_{\mathbf{x}}} = \mathbf{\Phi}\mathbf{m}_{2_{\mathbf{Q}}}\mathbf{\Phi}^{T} \tag{3.41}$$

or in scalar version

$$m_{2_{x_{ij}}}^{\texttt{CQC}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}\phi_{jm}m_{2_{Q_{mm}}} + \sum_{m=1}^{\texttt{NM}} \sum_{n=1,n\neq m}^{\texttt{NM}} \phi_{im}\phi_{jn}m_{2_{Q_{mn}}} \tag{3.42}$$

and since in general only variances (i.e. auto-covariances) of structural responses are of interest, Equation (3.42) reduces to

$$m_{2_{x_i}}^{\texttt{CQC}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}^2 m_{2_{Q_{mm}}} + \sum_{m=1}^{\texttt{NM}} \sum_{n=1,n\neq m}^{\texttt{NM}} \phi_{im}\phi_{in}m_{2_{Q_{mn}}} \tag{3.43}$$

where $m_{2_{x_i}}$ denotes the variance of the i-th structural degree-of-freedom's response.

Both Equation (3.42) and Equation (3.43) are the sum of two contributions: (i) the contribution of modal variances (first sum), and (ii) the contribution of the modal covariances to the structural responses (second term). When both terms are considered, they represent the so-called *Complete Quadratic Combination* (CQC) of modal responses. While this indeed introduces an higher level of complexity as well as numerical burden, it is the most accurate approach in the determination of the second order statistics of resulting structural responses.

Nonetheless, a simplified approach is available for the determination of statistics of structural responses from the ones of the modal responses. It takes the name of *Square Root of the Sum of the Squares* (SRSS), where statistics of the resulting structural responses are obtained as a linear combination of auto-covariance of modal responses only, that is

$$m_{2_{x_i}}^{\texttt{SRSS}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}^2 m_{2_{Q_{mm}}} \tag{3.44}$$

which is clearly equal to considering only the first term in Equation (3.43).

3.3.3 Third order

Once the bispectra of nodal loads are determined, assuming to conduct the analysis in the modal basis (the advantages of adopting such an approach have been discussed in the introductory part), the bispectra of the resulting modal forces are directly defined as (see Equation (3.31))

$$B_{P_{mno}}(\omega_1, \omega_2) = \sum_{i=1}^{\texttt{NDOFs}} \sum_{j=1}^{\texttt{NDOFs}} \sum_{k=1}^{\texttt{NDOFs}} \phi_{im}\phi_{jn}\phi_{ko} B_{f_{ijk}}(\omega_1, \omega_2) \tag{3.45}$$

where ϕ_{im} is an element of the modal matrix $\mathbf{\Phi}$ (i.e. the eigenvectors).

Then, in the same spirit of what has been seen at second statistical order, the formulation of the third order stochastic dynamic equation reads:

$$B_{Q_{mno}}(\omega_1, \omega_2) = K_{2_{mno}}(\omega_1, \omega_2) B_{P_{mno}}(\omega_1, \omega_2) \tag{3.46}$$

where

$$K_{2_{mno}}(\omega_1, \omega_2) = H_m^\star(\omega_1) H_n^\star(\omega_2) \overline{H_o^\star}(\omega_1 + \omega_2) \tag{3.47}$$

represents the second Volterra Kernel. It is called *second* Volterra Kernel because, in the time domain, for stationary processes it is function of two independent time lags τ_1, τ_2 only.

Then, following Equation (2.45), the third order moments of the resulting modal responses are the results of the double integration in the 2D frequency space of the full 3D tensor matrix of bispectra of modal responses:

$$m_{3_{Q_{mno}}} = \iint_{-\infty}^{\infty} B_{Q_{mno}}(\omega_1, \omega_2) d\omega_1 d\omega_2 \tag{3.48}$$

Then, similarly to second order, also third moments of structural responses are given by linear combination of the respective modal responses:

$$m_{3_{x_{ijk}}}^{\texttt{CCC}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}\phi_{jm}\phi_{km} m_{3_{Q_{mmm}}} + \sum_{m=1}^{\texttt{NM}} \sum_{n=1}^{\texttt{NM}} \sum_{\substack{o=1 \\ (n,o)\neq(m,m)}}^{\texttt{NM}} \phi_{im}\phi_{jn}\phi_{ko} m_{3_{Q_{mno}}} \tag{3.49}$$

and, focusing on auto-moments only, Equation (3.49) becomes

$$m_{3_{x_i}}^{\texttt{CCC}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}^3 m_{3_{Q_{mmm}}} + \sum_{m=1}^{\texttt{NM}} \sum_{n=1}^{\texttt{NM}} \sum_{\substack{o=1 \\ (n,o)\neq(m,m)}}^{\texttt{NM}} \phi_{im}\phi_{in}\phi_{io} m_{3_{Q_{mno}}} \tag{3.50}$$

Both Equations (3.49) and (3.50) refer to the so-called *Complete Cubic Combination* (CCC, along the same line of the well known CQC). Also in this case, the simplified approach which considers auto- modal responses only is applicable. It takes the name of *Cubic Root of the Sum of the Cubes* (CRSC), and reformulates Equation (3.50) as

$$m_{3_{x_i}}^{\texttt{CRSC}} = \sum_{m=1}^{\texttt{NM}} \phi_{im}^3 m_{3_{Q_{mmm}}} \tag{3.51}$$

which consists in considering only the first sum in Equation (3.50).

One important remark has to be done at this point. The SRSS (simplified second order) combination of modal responses has been widely adopted, specially in the past, when computing power was nothing close to that of today's machines. Nonetheless, it has been employed even up to recent years, since for what concerned second order statistical moments (i.e. variances and covariances), the discrepancies in resulting statistics of structural responses (see Equations (3.42) and (3.44)) was (generally assumed) negligible. Some authors [39, 40] have already proved this common practice to be harmful (or uneconomical) in some cases, showing that the two approaches can indeed lead to different results. However, it is nonetheless an acceptable assumption in most cases. The same cannot be said when considering higher order statistics. In fact, if at second order, it can be proved mathematically that in the 2D matrix of covariances of a given random process, covariances (outer-diagonal elements) are almost always smaller (in absolute value)[1] with respect to variances, the same does not hold anymore once considering higher order statistics (i.e. higher than second order). As a consequence, CRSC (more than SRSS at second order) method is not recommended. This implies the need to accurately estimate a proportionally large number of outer-diagonal elements of a 3D matrix, rather than in a 2D case. Examples will follow showing this aspect.

[1] This comes from the property that the correlation coefficients are scalar numbers at most equal to unity, in absolute value, that is $\rho = \in [-1, 1]$.

Chapter 4

Non-Gaussian buffeting analysis

This Chapter will focus on the wind loading. First, an overview of current practices will be provided. Then, a generalisation of the wind forces model will be formulated, introducing the concept of *elementary wind components*, as a base for computing the extended set of wind components. Then, a specialisation of such generalised model for Bispectral analyses will be detailed. The final objective of this Chapter is: for a given FE structural model, provide the CPSD and Cross-Bispectra matrices of wind loads.

4.1 Properties of the wind load

As a dynamic natural action, wind is a very complex physical phenomenon. It involves many fields of study, such as fluid dynamics and meteorology, among others [41]. Wind is a fluid flow generated by the temperature gradients in the atmosphere, caused by a difference in solar heating of the Earth's surface [42, 43]. Hence, there are infinite number of atmospherical configurations, each one of them causing a unique turbulent wind flow within the atmospheric boundary layer (ABL) (see for example [42] for detailed information). Figure 4.1 shows a schematic representation of such boundary layer.

Being a fluid flow, wind is mainly characterised by its velocity, which together with some other properties of the fluid (i.e. atmospheric air) such as density, dynamic and kinematic viscosities, are at the base of the mathematical equations that characterise viscous fluids, better known as the Navier-Stokes equations. As it will also be briefly discussed in a later Section, Navier-Stokes equations are at the core of fields such as Computational Fluid Dynamics (CFD) [44, 45], where in Wind Engineering is often referred to as Computational Wind Engineering (CWE) [46, 47].

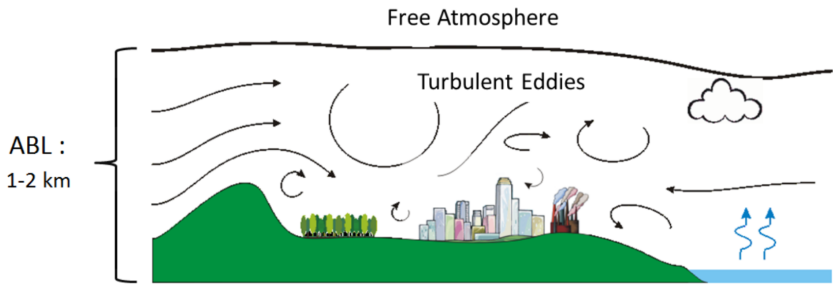


Figure 4.1: Schematic representation of the Atmospheric Boundary Layer (ABL).

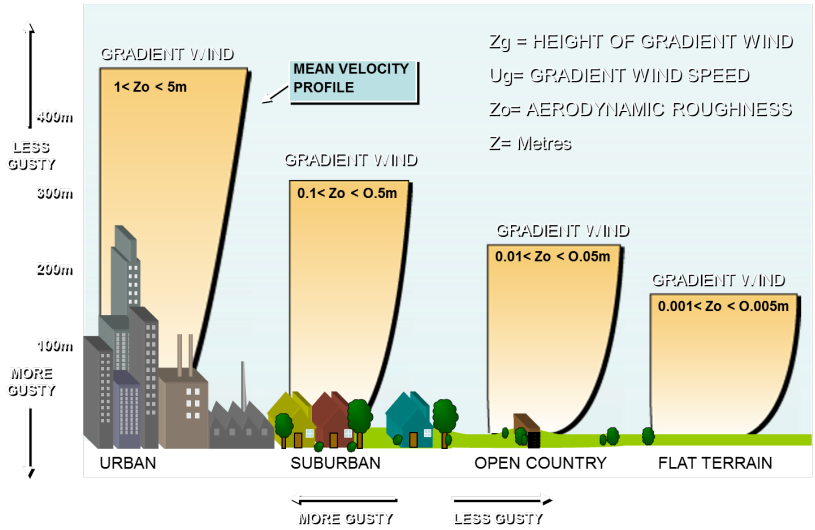


Figure 4.2: Example of terrain category exposure levels, in the ABL. Source: https://cppwind.com/wind-profile-characterization/

Under a pure engineering point of view [48], wind flow mean velocity characterisation is done based on what's called the *terrain category*, referring to a specific local terrain morphology and topography [49].

Figure 4.2 shows a graphical representation of such categorisation. The figure extends in two directions: (i) along the vertical axis, referring to the altitude, the wind velocity increases, mainly due to meteorological macro-effects (atmospheric flows); (ii) along the horizontal axis, moving from left to right, the wind velocity increases (at the same height) due to *undisturbed* flow conditions, where the fluid flow is not interrupted or altered by any obstacle, keeping its kinematic energy. This is contrary to what happens in urban areas (left-most side), where the high concentration of buildings acts as a barrier to the natural wind flow.

In practice, there are two main empirical laws to compute the mean wind velocity at a given altitude $\overline{U}$: (i) power law and (ii) logarithmic law:

$$\overline{U}(z) = \left(\frac{z}{z_{\text{ref}}}\right)^{\alpha}; \quad \overline{U}(z) = \frac{1}{k}\sqrt{\frac{\tau_0}{\rho}}\ln\frac{z}{z_0}. \tag{4.1}$$

When dealing with turbulent flows, such as wind flow in the ABL, it is convenient to determine a local reference system (Wind Reference System, WRS in this specific case), where at least one normal base vector of such reference system is parallel to the mean flow. For the case of the WRS, it is common practice to align the X_{WRS} local axis to the mean flow direction, so that at each point (x, y, z) in space, the local velocity vector can be split into three Cartesian components:

$$V(t) = \begin{Bmatrix} U_{\text{X}_{\text{WRS}}}(t) \\ U_{\text{Y}_{\text{WRS}}}(t) \\ U_{\text{Z}_{\text{WRS}}}(t) \end{Bmatrix} = \begin{Bmatrix} \overline{U} + u(t) \\ v(t) \\ w(t) \end{Bmatrix}. \tag{4.2}$$

Therefore, if $\overline{U}$ is the mean wind velocity component, $(u(t), v(t), w(t))$ are the three spatial components of the turbulent part of the wind velocity. In Equation (4.2) direct dependence on spatial coordinates (x, y, z) of any wind velocity component has been dropped for conciseness in notations.

In the following, the set $(u(t), v(t), w(t))$ of wind velocity turbulent components will be referred to as set of *elementary turbulent components*. This nomenclature is adopted, as it will be clearer in the following sections, to distinguish this base information from the set of wind components resulting from an arbitrary wind model used for determining wind forces (or pressures).

Wind turbulent components are random by nature. It is well establish to consider them as random Gaussian processes, having zero mean. Moreover, for civil engineering purposes, usually wind action is usually defined on 10-minute windows [48]. *Turbulence intensity* is defined as

$$I_u = \frac{\sigma_u}{\overline{U}} \qquad I_v = \frac{\sigma_v}{\overline{U}} \qquad I_w = \frac{\sigma_w}{\overline{U}} \tag{4.3}$$

where σ_ξ is the standard deviation of the turbulent component in the $\xi-$th direction (x, y, z), in the WRS.

When it comes to model the spatial distribution of wind, two other important properties are defined for each wind component $v \in \{u, v, w\}$: the matrices of Cross-Correlations $\mathbf{R}_v(\tau)$ and Cross-PSDs $\mathbf{S}_v(\omega)$.

In a time domain approach, the Cross-Correlation of wind turbulent components evaluated at two spatial points I and J is expressed as:

$$R_{v_I v_J}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} v_I(t) v_J(t+\tau) dt. \tag{4.4}$$

Note that in Equation (4.4), mean values are omitted since wind turbulence is a zero-mean random process.

On the other hand, the cross-PSDs of wind turbulent components evaluated at two points I and J are expressed as

$$S_{v_I v_J}(\omega) = \Gamma_{v_{IJ}}(\omega) \sqrt{S_{v_I}(\omega) S_{v_J}(\omega)} \tag{4.5}$$

where $\Gamma_{v_{IJ}}(\omega)$ is the spatial coherence function of the wind turbulent component $v(t)$.

In practice coherence is determined via site measurements. Turbulence is measured at points I and J, and coherence is inferred through Cross-PSD of long measured time series.

As one might compute the cross-spectrum of a given turbulent component at two different points in space, one could also compute the same information for two different turbulent components [50, 51], say $u(t)$ and $w(t)$ at either (i) the same point in space, or more generally (ii) at two different spatial locations. While this aspect is indeed of crucial importance, yet little is known about the actual coherence between different turbulent components at different points in space.

From a modelling point of view, current models for the cross coherence functions are usually (simply) decreasing exponential functions [2], although nothing justifies this choice from a physical standpoint [52]. It is recognised [52] that the current models are not generic enough and should be significantly adapted, from site to site.

For these reasons, while the author acknowledges the need of specific meteorological studies to provide a better understanding of this phenomenon, in this work, any possible correlation existing between different turbulent components is discarded, as a neutrality choice: if on the one hand, this might signifies that an important physical aspect is neglected, on the other, the introduction of (yet) unquantifiable errors due to the lack of extensive studies able at providing reliable models is avoided.

4.2 The quasi-steady aeroelastic modelling in Wind Engineering

Before proceeding to the details of the mathematical aspects of the buffeting problem, two important concepts must be addressed, aerodynamic and aeroelastic.

Fung in [53] defines aeroelasticity as:

> "the study of the effect of aerodynamic forces on elastic bodies."

However, the specificities of such statement are much more complicated. In fact, in this context, one major classification can be (and is practically) done: (i) cases where the aerodynamic wind forces do not depend on the body motion, (ii) cases in which, instead, the body motion affects the determination of the external loading. In the latter case, the wind forces are a function of the body motion, which assumed the general equations of motion of a MDOF system (see Equation (3.1)), means that the appplied wind forces are a function of the structural displacements and velocities, i.e. $\mathbf{f}(t) \equiv g(\mathbf{x}(t), \dot{\mathbf{x}}(t))$. These configuration-dependent aerodynamic forces are often referred to as *self-exciting* wind forces component.

There exist several physical phenomena that are studied considering these self-exciting forces:

- the *flutter* instability phenomena, affecting historically mostly airplane wings [54]. Now, in reality, a problem to be also accounted when designing very slender structures, such as long span bridges. Very famous the collapse of the Tacoma Bridge in 1940 [55, 56, 57, 58].
- *vortex induced vibrations*, which, as the name suggests, are vibrations induced by the alternating wake generated by the flow around a bluff body [59].

Contrarily, if the structural motion can be considered slow enough compared to the flow speed, so that it does not affect the wind flow around the bluff body, it is safe to neglect these self-exciting forces, and so consider that the global aerodynamic forces acting on the structural system depend solely on the temporal (and spatial) configuration of the incoming flow, relative to the structural configuration. Aeroelastic problems where self-exciting forces are discarded are sometimes referred to as simply *aerodynamic* problems, since the final loading depends only on the aerodynamic features of both wind flow and structural shapes, without accounting of any possible fluid-structure interaction. This naming can be justified to make clear the context, i.e. the assumption of undisturbed flow and so consequent drop of any self-exciting aeroelastic effect.

If unstable aeroelastic effects (e.g. those including unsteady self-exciting aeroelastic forces, which are usually responsible of the destabilisation of the system at a critical configuration) are to be avoided for safety reasons, buffeting effects are to be controlled for human comfort and serviceability of the structural system, but also for safety at the Ultimate Limit States (ULS) for which the structure must be designed for following Standards requirements [48].

In the following, the assumption of undisturbed wind flow will be made, so that to discard any self-exciting aeroelastic effects, and focus mainly on the buffeting problem is allowed.

4.3 Wind buffeting forces & the set of *elementary* wind components

When focus is made only on the family of quasi-steady aeroelastic models, resulting applied loads are obtained as memoryless transformations of the set of wind turbulent components.

There exist several models to express the aerodynamic wind loads at DOF i of node I of a finite element model, as a function of the average wind velocity $\overline{U}_I$ at that node and $u(t), v(t), w(t)$, the elementary velocity components of wind turbulence, at the same node only. Other more complex models can also express the aerodynamic load at a given node I as the result of the interaction of the flow turbulence along the entirety of the structural system, hence accounting for possible interference specially with closely-spaced points. Details will be given in the following.

In the following sections, a first discussion and overview of currently adopted aerodynamic models will be given. Not only to present current practices, but also to prepare the field for a more general formalisation of aerodynamic models, hence accounting for non-Gaussian intrinsic nature of aerodynamic loading. Such generalised model is at the base of the new mathematical and numerical developments discussed in this Thesis.

When dealing with buffeting loads on a line-like structure (i.e. bridge decks, towers), three main forces can be defined:

- Drag (D) force, parallel to the mean wind flow direction;
- Lift (L) force, transversal to the mean wind flow direction;
- Pitching moment (M), turning moment around an axis perpendicular to both the wind flow and the lift direction vector, by convention positive when it tends to raise the leading edge of the body.

Mathematically, these aerodynamic forces are expressed as:

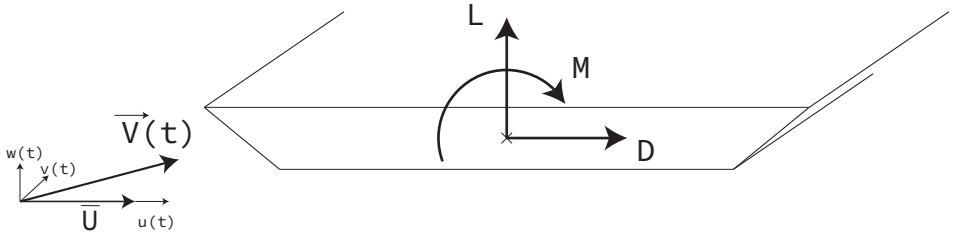


Figure 4.3: Schematic representation of the wind velocity vector (composed from the mean and *elementary* wind turbulent components) acting on a bridge deck, generating the 3 buffeting forces: drag, lift, and pitching moment.

$$\begin{aligned} F_D(t) &= \frac{1}{2}\rho C_D(i(t))BV^2(t) \\ F_L(t) &= \frac{1}{2}\rho C_L(i(t))BV^2(t) \\ M(t) &= \frac{1}{2}\rho C_M(i(t))B^2V^2(t) \end{aligned} \tag{4.6}$$

where ρ is the air density, C_D, C_L, C_M the drag, lift, moment aerodynamic coefficients, B the characteristic length of the bluff body,

$$V^2 = ||\overrightarrow{V}||^2 = (\overline{U} + u(t))^2 + v^2(t) + w^2(t) \tag{4.7}$$

the square of the norm of the instantaneous wind velocity vector, as shown in Figure 4.3. It is composed of three spatial components

$$\overrightarrow{V}(t) = \begin{Bmatrix} \overline{U} + u(t) \\ v(t) \\ w(t) \end{Bmatrix} \tag{4.8}$$

when expressed in the WRS, where the X axis is taken parallel to the mean wind flow direction. This set of spatial turbulence components is at the base of the following developments. Therefore, it will be referred to as the *elementary* set of turbulence components, or shortly *elementary* wind components.

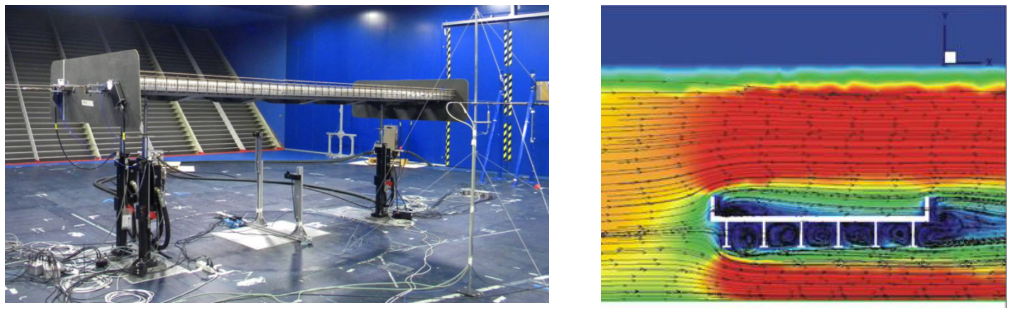
The aerodynamic coefficients C_D, C_L, C_M are obtained by rewriting Equations (4.6) as

$$C_D(t) = \frac{F_D(t)}{\frac{1}{2}\rho BV(t)^2} \quad ; \quad C_L(t) = \frac{F_L(t)}{\frac{1}{2}\rho BV(t)^2} \quad ; \quad C_M(t) = \frac{F_M(t)}{\frac{1}{2}\rho B^2V(t)^2}. \tag{4.9}$$

Aerodynamic forces F_D, F_L, M are computed or measured in a Wind Tunnel and this allows for an estimate of the structural aerodynamic coefficients C_D, C_L and

C_M [60], which might then be used in a design procedure, to compute effective aerodynamic forces, based on an accurate characterisation of the effective wind flow.

Alongside, in the last decade, the numerical approach aimed at directly solving the Navier-Stokes equations (Computational Fluid Dynamics (CFD), also named Computational Wind Engineering (CWE) in such context) is increasingly used for this purpose of estimating the aerodynamic coefficients, specially for very complex body shapes, for which manufacturing scaled models can be a challenge. In that, CWE is relatively cheap, since it only requires good computing power in order to be operational. Nonetheless, for large simulations, the required computing power might be considerable, such that its costs might be significant as well.



(a) (b)

Figure 4.4: Examples of wind forces estimation:
(a) Wind-Tunnel setup (Source[1]), (b) CFD model (Source[2]).

4.4 The intrinsic Non-Gaussian nature of the aerodynamic load and the dimensionality problem

The study of wind effects on structures involves the combination of structural dynamics and loading models, concepts introduced in Sections 3.1 and 4.2.

As for the loading model, very general and complex formulations have been formulated [61, 62], which consider the wind flow inclination and orientation in a 3D space. In this Thesis, only the so called *normal* winds will be considered. So that, referring to the general nomenclature (see [62] for instance), the yaw angle will be always 0, and the only wind incidence is the one lying in the cross-sectional plane. This is represented in Figure 4.5.

[1] https://www.windtunnel.polimi.it/bridge-aerodynamics-and-aeroelasticity/
[2] https://www.fhwa.dot.gov/publications/research/infrastructure/hydraulics/09028/004.cfm

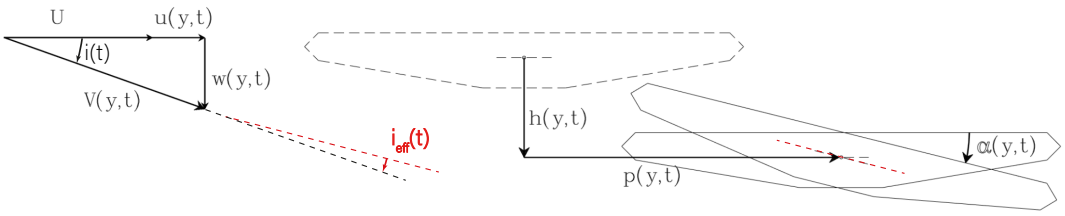


Figure 4.5: Wind turbulence incidence angle. With structure still (dashed line), and with structure moving [5].

Figure 4.5 also shows a schematic diagram where the elastic body has moved from its initial (undisturbed, dashed line) configuration along all the degrees-of-freedom in its plane, i.e. two in-plane displacements, vertical and horizontal, and a rotation around the axis normal to the plane.

Such displaced configuration affects both the determination of the effective incidence, as well as the effective norm of the wind velocity vector (see Equation (4.8)):

$$i_{eff}(t) = \arctan\left[\frac{w(t) - \dot{h}(t) - rB\dot{\alpha}(t)}{\overline{U} + u(t) - \dot{p}(t)}\right] - \alpha(t) \tag{4.10}$$

$$\begin{aligned} {V_{eff}}^2 &= (\overline{U} + u(t) - \dot{p})^2 + v(t)^2 + (w(t) - (\dot{h} + rB\dot{\alpha}(t))^2 \\ &= \overline{U}^2 + u^2 + \dot{p}^2 + 2\overline{U}u - 2\overline{U}\dot{p} - 2u\dot{p} + v^2 \\ &\quad w^2 + \dot{h}^2 + (rB\dot{\alpha})^2 - 2w\dot{h} - 2wrB\dot{\alpha} + 2\dot{h}rB\dot{\alpha} \end{aligned} \tag{4.11}$$

so that, generally, also the quasi-steady buffeting loading dependent on the instantaneous system configuration, results in a nonlinear transformation of $u(t)$, $v(t)$ and $w(t)$.

It is widely accepted that the elementary wind turbulence components are assumed to be Gaussian random processes [28, 63]. On the other hand, aerodynamic pressures and forces are in principle non-Gaussian [64, 65]. Inserting Equations (4.10) and (4.11) into the formulation of aerodynamic wind forces given in Equation (4.6) is what introduced the non-linearity and so the intrinsic non-Gaussian nature of aerodynamic forces, due to the presence of powers higher than 1 of turbulence components. In fact, squaring a Gaussian random process results in a non-Gaussian process, since squaring is a nonlinear operator. So that, even though the elementary wind turbulent components (u, v, w) are assumed to be Gaussian random processes, the resulting wind forces do not necessarily have the same property.

Besides, with aid of advancing measurement devices, there have been evidences which showed that measured wind turbulence in the Atmospheric Boundary Layer (ABL) does not strictly follow a Gaussian distribution [63]. However, these results were mainly obtained in measurements of exceptional winds scenarios, such as typhoons and hurricanes, which fall outside the conditions in which buffeting problem, treated in this context, may occur. Nonetheless, recently Li et al. in [66] showed that even synoptic wind turbulence, category in which buffeting winds fall, might show non-Gaussianity due to the weak stationarity of the randomly fluctuating wind turbulence velocities.

Other works [10] also tackle the problem of non-Gaussian turbulence in other domains of application. For clarity, the possibility of non-Gaussian turbulence is not explicitly considered in the presented work, so that elementary wind turbulence velocity components are assumed to be random and Gaussian. This is because a full aerodynamic model encompassing wind pressures and wind loads is provided, and how PSDs and bispectra of wind loads can be obtained from wind velocities is explained. In other words, a mathematical model based on the quasi-steady approximation is developed, expressing other stochastic propoerties of the non-Gaussian wind loads as a function of Gaussian turbulence. Alternatively, any other loading model, e.g. with non-Gaussian wind turbulence, could be studied too, provided that finally the PSDs and bispectra of wind loads are obtained.

It is important to note that non-Gaussianity does not mean non-linearity of the structural system, so that, a frequency domain approach to the solution of equations of motion in Equations (3.1) or (3.2) is still valid. Hence, for the frequency domain approach to the dynamic problem, the loading need not to be necessarily Gaussian, as some authors have pointed out [11].

While in this work bispectral analysis will be applied in frameworks where the non-Gaussianity comes from the input, applied to a linear dynamic system (by means of some nonlinear transformations), some authors have used bispectral analysis to identify light non-linearities of systems excited by some random Gaussian white noise [67, 68, 16]. Other authors have extended these applications to include stationary non-Gaussian inputs as well [69]. In [70], authors have investigated the stochastic response of a beam, assuming both linear and nonlinear mechanical behaviours, under a non-Gaussian random excitation.

In the general case, accounting for the non-Gaussian nature of aerodynamic forces brings a novel degree of complexity to the problem, for both time and frequency domain.

If non-Gaussian properties are to be accounted when dealing with aerodynamic forces (as it should per its natural definition), care has to be used in properly characterising the underlying non-Gaussian process, so to avoid any statistical misrepresentation. And the need of dealing with very large time series,

or spectra, hence calling for high-dimensional quantities, must be acknowledged. This in part also explains why, up to current times, no real applications in which this aspect has been concretely accounted for are found, at least to the author's knowledge [71].

This Thesis focuses on the non-Gaussian buffeting problem. This is done with the final goal of firstly determining, and secondly proving, that neglecting this important feature of the aerodynamic loading might potentially affect statistics of the resulting structural responses, causing either (i) unsafe or (ii) uneconomical design solutions. In fact, if the loading process is non-Gaussian, the resulting structural response could potentially be non-Gaussian as well, to the same degree. However, there might actually be a second scenario where accountancy of non-Gaussianity might play an important role in the problem under consideration. This could include the aeroelastic phenomena introduced at the beginning of this section. Clearly, in such cases, the response tends to a Gaussian harmonic response, since the structure starts vibrating regularly at the frequency of the targeted modal motion (depending on the effect). However, neglecting non-Gaussianity of the aerodynamic loading only for the reason that it is expected the response to be pure Gaussian, might in reality bias the effective energetic content [6] (better visualised in the frequency domain) of the load, which might finally affect the determination of the critical condition at which these unstable effects are expected to happen.

4.5 Wind model: a general approach

The concept of *wind model* plays a fundamental role in the context of this Thesis. A wind model is any mathematical relation that links the spatial wind field, expressed in terms of velocities $(\overline{U}+u(t), v(t), w(t))$, to aerodynamic forces $(f(t))$ that are applied on a body immersed in the field.

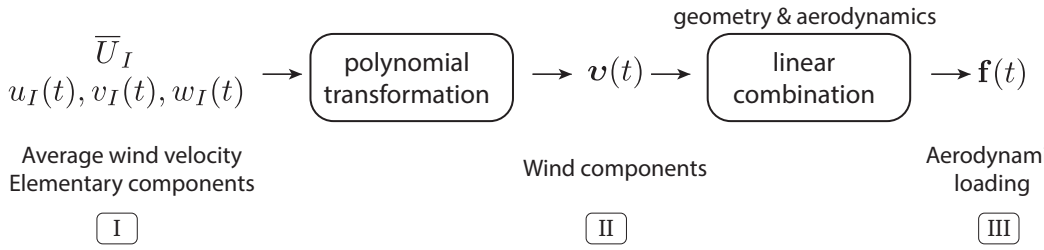


Figure 4.6: Schematic representation of the general nonlinear polynomial aerodynamic loading model.

Figure 4.6 shows a schematic representation of a wind model. It consists of two steps:

1. From *elementary* wind components $(\overline{U} + u(t), v(t), w(t))$, which constitute the three spatial components of the wind field at any given point in space, to wind components $\boldsymbol{v}(t)$ that represent the result of the basic interacting components in the turbulent flow.

2. From wind components $\boldsymbol{v}(t)$ to aerodynamic forces $f(t)$.

While the second step which translates wind components to aerodynamic forces is quite restrained, since it is heavily dependent on the aerodynamic features of the body immersed in the wind flow, the first step that transforms the *elementary* wind components into the final set of wind components is the primary source of differentiation between wind models.

In essence, the first step is what establishes the *complexity* of a given wind model. For instance, if the polynomial transformation in Step I-II is chosen to obey to a linear relation, since the operations linked to step II-III in Figure 4.6 are also linear, then the overall mathematical relation between wind turbulence and aerodynamic forces will be linear as well. On the contrary, any non-linear relation that will reside in Step I-II, will be inevitably reflected in Step II-III, resulting in wind forces that are a linear combination of possibly non-linear wind components.

4.5.1 Linearisation of the buffeting forces & the linearised quasi-steady aerodynamic wind model

Whenever a complex relation exists between input and output of a given problem, the very first thing that is usually tried is to check whether any kind of simplification can be employed. Linearising a nonlinear problem is indeed one of such approaches. In fact, linear relations are the most simple to be solved, at least as a first approximation to understand the system's response to a given input.

Because of its intrinsic non-linearity, the aerodynamic loading is usually linearised.

In this regard, very few records of cases where the non-Gaussian nature of the wind buffeting loading has been accounted are found in the literature, specially not considering only simple SDOF systems. Although some would justify this linearisation approach by stating that the physical conditions would allow for such simplification without distorting too much the (yet approximate) representation of reality, the main reason is indeed its mathematical formulation that is simple and easy to handle, even in the most complex applications. In this regard, some work has been done by Gusella and Materazzi in [12] and [8], where the authors mainly focused on comparing the along-wind response of a tall and line-like structure in the two scenarios, i.e. with and without the Gaussian loading assumption. Gioffrè and Gusella in [70] studied the stochastic response of

a system under non-Gaussian wind loading, considering both scenarios of linear and nonlinear mechanical behaviour.

Some authors did consider this non-linearity, but only in time domain applications [72, 7].

Linearisation of buffeting forces requires:

- Linearisation of the nonlinear *arctan* operator in the formulation of effective incidence angle in Equation (4.10);
- Any quadratic term, as well as any term coupling turbulence and structural velocities, to be neglected in the formulation of the effective wind velocity norm in Equation (4.11).

This requires the formulation of two important assumptions:

- Turbulence velocities are small compared to the mean wind flow speed, which is equivalent to having low turbulence intensities (see Equation (4.3));
- Induced rotations and velocities are small, so that their effect on the oncoming flow can be neglected.

These lead to

$$i_{eff\_lin}(t) = \frac{w(t)}{\overline{U}} \tag{4.12}$$

$$V_{eff\_lin}{}^2 = \overline{U}^2 + 2\overline{U}u - 2\overline{U}\dot{p}. \tag{4.13}$$

Injecting Equations (4.12) and (4.13) in the definition of buffeting forces in Equation (4.6), and after some mathematical simplifications, the final linearised wind buffeting forces formulation reads:

$$\begin{aligned} F_D(t) &= \frac{1}{2}\rho B\overline{U}^2 c_d + \frac{1}{2}\rho B\overline{U}^2\left[2c_d\frac{u}{\overline{U}} + c_d{}'\frac{w}{\overline{U}}\right] \\ F_L(t) &= \frac{1}{2}\rho B\overline{U}^2 c_l + \frac{1}{2}\rho B\overline{U}^2\left[2c_l\frac{u}{\overline{U}} + c_l{}'\frac{w}{\overline{U}}\right] \\ M(t) &= \frac{1}{2}\rho B^2\overline{U}^2 c_m + \frac{1}{2}\rho B^2\overline{U}^2\left[2c_m\frac{u}{\overline{U}} + c_m{}'\frac{w}{\overline{U}}\right] \end{aligned} \tag{4.14}$$

where $c_\xi = C_\xi(i_0)$ and $c_\xi{}' = \left.\frac{dC_\xi}{di}\right|_{i=i_0}$, $\xi = \{D, L, M\}$, i_0 being the mean incidence angle.

In a more compact form, Equation (4.14) can be rewritten as:

$$\begin{aligned} F_D(t) &= c_{d_0}\left[1+2\frac{u}{\overline{U}}\right]+c_{d_1}\frac{w}{\overline{U}} \\ F_L(t) &= c_{l_0}\left[1+2\frac{u}{\overline{U}}\right]+c_{l_1}\frac{w}{\overline{U}} \\ M(t) &= c_{m_0}\left[1+2\frac{u}{\overline{U}}\right]+c_{m_1}\frac{w}{\overline{U}} \end{aligned} \tag{4.15}$$

where $c_{\xi 0}=\frac{1}{2}\rho B^{\beta}\overline{U}^2 c_{\xi}$ and $c_{\xi 1}=\frac{1}{2}\rho B^{\beta}\overline{U}^2 c'_{\xi}$, $\beta=1$ for drag and lift, $\beta=2$ for moment.

From the formulation in Equation (4.15) it is evident how, in the linearised version of the buffeting forces, each force component depends only on a linear combination of unitary powers of the *elementary* wind turbulence components at most.

In most of today's real applications, *linear* wind load models are considered, particularly when a stochastic approach is adopted. When it comes to extending such linearised aerodynamic wind model formulation to MDOF systems, in the most general fashion the fluctuating component of the aerodynamic loads at all structural DOFs are given by

$$\mathbf{f}(t)=\mathbf{A}_0+\mathbf{A}^{(u)}\mathbf{u}(t)+\mathbf{A}^{(v)}\mathbf{v}(t)+\mathbf{A}^{(w)}\mathbf{w}(t) \tag{4.16}$$

where $\mathbf{f}(t)$ is the `NDOFs` $\times$ 1 vector of wind loads at all DOFs of the structural model, obtained as linear combinations of the wind components $\boldsymbol{v}(t)=\{\mathbf{u},\mathbf{v},\mathbf{w}\}$ at all aerodynamic nodes `NNa`. In such models, the set of wind components is composed of Gaussian processes only, and matches the set of *elementary* wind components. The wind components are linearly combined by means of *wind coefficients* (intrinsic to the wind transformation model and to the bluff body characteristics), embedded in the `NDOFs` $\times$ `NNa` matrices of wind coefficients $\mathbf{A}^{(\mu_k)}$ ($\mu_k \in \{u,v,w\}$, $k=\{1,2,3\}$). In the simplest case, $\mathbf{A}^{(u)}=\mathrm{diag}(c_{\xi 0}\frac{2}{\overline{U}})$ and $\mathbf{A}^{(w)}=\mathrm{diag}(c_{\xi 1}\frac{1}{\overline{U}})$ (see Equation (4.15)). However, they can assume more general forms in case of 3D structures and oblique or yawed wind flows. Wind coefficients are indeed the most important aspect of the quasi-steady family of wind models. Moreover, in the most general case the wind coefficients actually embed two transformation processes. Details are given in Appendix A.

Since the transformation from step II to III in Figure 4.6 is always linear, and since the *elementary* wind turbulence components $u(t)$, $v(t)$, $w(t)$ are assumed to be zero-mean Gaussian random processes [1], resulting aerodynamic wind forces expressed in Equation (4.15) will be Gaussian random processes as well [9].

As a consequence, the determination of the statistics of $\mathbf{f}(t)$ as a function of

those of $\boldsymbol{v}(t)$ is straightforward. In fact, the determination of the statistics of $\boldsymbol{v}(t)$ will automatically result in the determination of those of the aerodynamic wind loading, as a linear combination of them.

To summarise, Equation (4.16) can be reformulated as (see Equation (A.7))

$$\mathbf{f}(t) = \hat{\mathbf{X}} \left(\mathbf{a}^{(u)}\mathbf{u}(t) + \mathbf{a}^{(v)}\mathbf{v}(t) + \mathbf{a}^{(w)}\mathbf{w}(t) \right) \tag{4.17}$$

where $\hat{\mathbf{X}}$ is a $\texttt{NDOFs} \times \texttt{NNa}$ matrix which transforms wind forces applied at the discrete nodes to forces at all DOFs of the structural model.

By explicitly taking out the sum over the set of the loaded aerodynamic nodes NNa, Equation (4.17) reads

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \mathbf{x}_I \left(a_I^{(u)} u_I(t) + a_I^{(v)} v_I(t) + a_I^{(w)} w_I(t) \right) \tag{4.18}$$

where $\mathbf{x}_I$ if a $\texttt{NDOFs} \times 1$ vector translating the effect of the force applied at the I-th aerodynamic node ($I = 1, \ldots, \texttt{NNa}$) to the whole set of structural DOFs.

In practice, the vector $\mathbf{x}_I$ is mostly composed of zeroes since, in a finite element approach, the resulting loads at the different DOFs of a given node I are obtained by integration of the applied loads on elements connected to that node only. In addition to this, in many wind models, the loads at the DOFs of a given node of the finite elements model are expressed as a function of the components of wind velocity $\boldsymbol{v}_{I,:}(t)$ at the same node I (of the wind field)[3]. Or, in a more general way, wind loads at the level of the set of aerodynamic nodes NNa (resulting from the application of the wind model transformation) are the result of the wind flow (i.e. wind speeds) at the corresponding node. This is reflected by the fact that the wind coefficients $a_I^{(\mu_k)}$ in Equation (4.18) are scalar entities, meaning that they directly translate the wind field into wind force at that node I only. Some more advanced models could result in forces at a given (aerodynamic) node that are also expressed as a function of the wind velocity at neighboring nodes [73], in which cases Equation (4.18) would be reformulated as

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \mathbf{x}_I \left(\mathbf{a}_I^{(u)^T}\mathbf{u}(t) + \mathbf{a}_I^{(v)^T}\mathbf{v}(t) + \mathbf{a}_I^{(w)^T}\mathbf{w}(t) \right) \tag{4.19}$$

where now the wind coefficients $\mathbf{a}_I^{(\mu_k)}$ are $\texttt{NNa} \times 1$ vector entities which express the effect that the wind flow at all NNa aerodynamic nodes has on the aerodynamic node I. Additionally, by considering Equations (A.1), (A.2) and (A.3),

[3] This statement is true only under the assumption that aerodynamic and structural nodes are the same.

Equation (4.19) can be easily generalised to cases where aerodynamic nodes do not completely match structural ones.

4.5.2 A general polynomial formulation of the quasi-steady aerodynamic wind model

When referring to linear wind load model, it is meant that the smallness of the turbulence components with respect to the average wind velocity is considered, and a series expansion is developed to obtain a linear relation between turbulence components and the wind loads [74].

So, the assumption of small turbulence has been greatly welcomed since it allowed the full probabilistic description of the buffeting loading using the mean and variance only. However, as soon as quadratic components are included (i.e. $d > 1$) in the loading model as in Equation (4.27), some wind turbulence components such as $u^2(t)$ are no longer Gaussian, and so is the resulting loading.

In such a context, Equation (4.18), can be further generalised by expressing explicitly the sum over the set of wind turbulence components:

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \mathbf{x}_I \sum_{a=1}^{\texttt{NDEGW}} a_{Ia} \upsilon_{Ia}(t) \tag{4.20}$$

where $\texttt{NDEGW} = 3$ for now, or, in vectorial notation

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \mathbf{x}_I \mathbf{a}_I^T \boldsymbol{\upsilon}_I(t) \tag{4.21}$$

where $\boldsymbol{\upsilon}_I(t)$ represents the $\texttt{NDEGW} \times 1$ vector collecting the set of turbulent components acting at node I, $\mathbf{a}_I^T$ the vector of relative wind model coefficients (evaluated at the same I-th node).

The notation adopted in Equations (4.18) and (4.19) is at the same time general and efficient, since it resembles the localization operation in a standard finite element approach. To simplify notations, Equation (4.20) introduces $\upsilon_{Ia}(t)$ to refer to the a^{th} element in the full set of the NDEGW wind fluctuating component, at node I, where NDEGW refers to the number of elements in the set of turbulent components $\boldsymbol{\upsilon}_{I,:}(t)$. For instance, the first component in the set will refer to the wind turbulent component parallel to the direction of the mean wind $\overline{U}$, that is $\upsilon_{I1}(t) = u_I(t)$. Clearly, in the hypothesis of linearised wind model, the set of all possible wind turbulence components happens to coincide with the set of *elementary* wind components $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$.

Besides, Equation (4.20) is usually written in a different notation by moving the term $\mathbf{x}_I$ inside the sum over the number NDEGW in the set of wind turbulent

components

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \sum_{a=1}^{\texttt{NDEGW}} \mathbf{x}_I a_{Ia} \upsilon_{Ia}(t) = \sum_{I=1}^{\texttt{NNa}} \sum_{a=1}^{\texttt{NDEGW}} \mathbf{a}_{Ia} \upsilon_{Ia}(t) \tag{4.22}$$

where $\mathbf{a}_{Ia} = \mathbf{x}_I a_{Ia}$ collects the NDOFs wind loading coefficients translating the influence of the a^{th} fluctuating component at aerodynamic node I on the load at all DOFs. The summation on a, limited to NDEGW, is discussed in the sequel. In vectorial notation, Equation (4.22) becomes

$$\mathbf{f}(t) = \sum_{I=1}^{\texttt{NNa}} \mathbf{A}_I \boldsymbol{\upsilon}_I(t) \tag{4.23}$$

where

$$\mathbf{A}_I = \mathbf{x}_I \mathbf{a}_I^T \tag{4.24}$$

is the NDOFs $\times$ NDEGW matrix of wind coefficients accounting for the contribution of the entire set of wind turbulent components acting on node I $\boldsymbol{\upsilon}_I$ on the load at all structural DOFs.

In Equation (4.22) it is possible to see υ_{Ia} as an element of matrix $\boldsymbol{\upsilon}$ whose size is NNa $\times$ NDEGW. This matrix can be partitioned in two different ways:

$$\boldsymbol{\upsilon} = (\boldsymbol{\upsilon}_{:,1}, \boldsymbol{\upsilon}_{:,2}, \ldots, \boldsymbol{\upsilon}_{:,a}, \ldots, \boldsymbol{\upsilon}_{:,\texttt{NDEGW}}) = (\hat{\boldsymbol{\upsilon}}_1, \hat{\boldsymbol{\upsilon}}_2, \ldots, \hat{\boldsymbol{\upsilon}}_a, \ldots, \hat{\boldsymbol{\upsilon}}_{\texttt{NDEGW}}) \tag{4.25}$$

$$\boldsymbol{\upsilon} = \left(\boldsymbol{\upsilon}_{1,:}^T, \boldsymbol{\upsilon}_{2,:}^T, \ldots, \boldsymbol{\upsilon}_{I,:}^T, \ldots, \boldsymbol{\upsilon}_{\texttt{NN},:}^T\right)^T = (\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}_2, \ldots, \boldsymbol{\upsilon}_I, \ldots, \boldsymbol{\upsilon}_{\texttt{NN}}), \tag{4.26}$$

i.e. by lines or columns. The symbol $\boldsymbol{\upsilon}_{:,1}$ means all lines of the first column, while $\boldsymbol{\upsilon}_{1,:}^T$ all columns of the first line. This distinction has been also marked with the introduction of two different notations, $\hat{\boldsymbol{\upsilon}}_a = \boldsymbol{\upsilon}_{:,a}$, as well as $\boldsymbol{\upsilon}_I = \boldsymbol{\upsilon}_{I,:}^T$.

The wind loading coefficients $\mathbf{a}_{Ia}$ in Equation (4.22) are expressed as a function of the aerodynamic shape of the elements, as well as their orientation with the mean wind flow [75, 76, 62] (see Equation (4.6) for the basic example). In the following, no hypothesis is made on how $\mathbf{a}_{Ia}$ is constructed so that any memoryless wind loading model can be considered. Along the same line, a possible improvement of Equation (4.20) would be to use a low-pass filter corresponding to the so-called numerical admittance, related to the discretisation of the pressure field along elements into nodal values [73]. Although this is not included in the following developments, nothing hinders its consideration. Based on these arguments, the establishment of coefficients $\mathbf{a}_{i,I}$ is more or less involved, depending on the fact that the studied structure is a line-like structure [75], a large structure in a 2-D plane or even a large 3-D structure [77, 62].

In more general models, the wind loading can be replaced by a polynomial expression of any arbitrary degree d, after truncating the series after higher order terms. For instance, when the quadratic nature [78, 74] of the loading is taken

into account, $d = 2$, the set of wind components at a given node is extended to [41]

$$\boldsymbol{v}_I(t) = \Big(u_I(t), v_I(t), w_I(t), u_I^2(t), v_I^2(t), w_I^2(t), u_I(t)v_I(t), u_I(t)w_I(t), v_I(t)w_I(t)\Big) \tag{4.27}$$

where all possible combinations of the components of wind velocity up to second degree have been considered, for a total of $\texttt{NDEGW} = 9$ wind components. Higher degree models could also be envisaged by including turbulence loading components such as $u_I^3(t)$.

The number of new elements when adding the terms of degree d is given by $(d+2)(d+1)/2$ (3 new terms for $d = 1$, 6 new terms for $d = 2$, 10 new terms for $d = 3$, etc.) so that, in total,

$$\texttt{NDEGW} = \frac{1}{6}(d+1)(d+2)(d+3) - 1 \tag{4.28}$$

if terms up to degree d are kept.

All in all, in the most generic format, the aerodynamic loads on the structure are therefore expressed by Equation (4.22), or Equation (4.23) in its vectorial form, where summations are performed on all aerodynamically loaded nodes (I) and all components of the wind load (a), which could be more than just three in case of nonlinear aerodynamic models. Besides, by choosing to embed all possible combinations of the various degrees of the elementary components, a very general meaning to Equations (4.22) and (4.23) is given.

Indeed, although $\mathbf{f}(t)$ is written as a linear transformation of $\boldsymbol{v}(t)$, it is possible to include nonlinear memoryless transformations of the turbulence components up to any desired order.

This model extends the linear formulation used by the scientific community so far. This expression is however significantly different: it conserves the properties of linear combinations (step II to III in Fig. 4.6), but of wind components that are themselves obtained as (possibly nonlinear) polynomial transformations of the elementary turbulence components (step I to II in Fig. 4.6) [41].

4.6 Generalised wind model: a stochastic "block" representation

When the stochastic approach is applied to the buffeting analysis, the main objective is to derive the statistics of $\boldsymbol{v}(t)$, knowing those of the wind velocities $\mathbf{u}(t)$, $\mathbf{v}(t)$ and $\mathbf{w}(t)$ (e.g. the set of *elementary* wind turbulence components, at all aerodynamic nodes), which are assumed to be zero-mean random Gaussian

processes. In other words, the main goal is to determine the complete probabilistic description of all wind components of $\boldsymbol{\upsilon}(t)$ (see Equations (4.25) and (4.26)) expressed as a function of the statistical description of $u_I(t)$, $v_I(t)$ and $w_I(t)$ at all aerodynamic nodes `NNa`.

So that, once probabilistic description of the buffeting load is available, relative statistical descriptors of the resulting structural responses can be easily obtained solving the stochastic version of the general equation of motion of MDOF systems (see Equations (3.39) and (3.46)). Besides, since the elementary wind components are assumed to be zero-mean, stationary, ergodic random Gaussian processes, this results in relating the probabilistic description of all wind components in $\boldsymbol{\upsilon}(t)$ to the (cross-)correlation functions of $u_I(t)$, $v_I(t)$ and $w_I(t)$ or, alternatively, of their (cross-)power spectral densities. This allows then to easily find the statistics of the resulting wind buffeting loading $\mathbf{f}(t)$ in Equation (3.1), and finally those of the structural response $\mathbf{x}(t)$.

In particular, considering the block representation in Equation (4.25), the (cross-)correlation matrix of the wind components takes the form

$$\mathbf{R}_{\hat{\boldsymbol{\upsilon}}}(\tau) = \begin{pmatrix} \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{11}}(\tau) & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{12}}(\tau) & \cdots & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{1,\mathtt{NDEGW}}}(\tau) \\ \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{21}}(\tau) & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{22}}(\tau) & \cdots & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{2,\mathtt{NDEGW}}}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{\mathtt{NDEGW},1}}(\tau) & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{\mathtt{NDEGW},2}}(\tau) & \cdots & \mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{\mathtt{NDEGW},\mathtt{NDEGW}}}(\tau) \end{pmatrix} \tag{4.29}$$

where $\mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{ab}}(\tau)$ represents the $\mathtt{NNa} \times \mathtt{NNa}$ (cross-)correlation matrix of wind components $\hat{\boldsymbol{\upsilon}}_a$ and $\hat{\boldsymbol{\upsilon}}_b$ (evaluated at all aerodynamic nodes), that is

$$\mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{ab}}(\tau) = \mathbb{E}\left[\hat{\boldsymbol{\upsilon}}_a(t)\hat{\boldsymbol{\upsilon}}_b^T(t+\tau)\right] \tag{4.30}$$

or in its scalar version

$$R_{\upsilon_{aIbJ}}(\tau) = \mathbb{E}\left[\upsilon_{aI}(t)\upsilon_{bJ}(t+\tau)\right] \tag{4.31}$$

where $R_{\upsilon_{aIbJ}}(\tau)$ represents the (cross-)correlation between wind components a and b at aerodynamic nodes I and J respectively.

It is noticed that the expressions of $\mathbf{R}_{\hat{\boldsymbol{\upsilon}}_{ab}}(\tau)$ can be obtained as a function of $\mathbf{R}_u(\tau)$, $\mathbf{R}_v(\tau)$ and $\mathbf{R}_w(\tau)$ no matter the degree of the polynomial transformation used to express $\hat{\boldsymbol{\upsilon}}_a$ and $\hat{\boldsymbol{\upsilon}}_b$ in terms of powers of u, v, w [4]. Detailed examples showing this aspect will be given in the following. In the frequency domain, the

side-by-side Fourier transform of Equation (4.29) gives

$$\mathbf{S}_{\hat{\boldsymbol{v}}}(\omega) = \begin{pmatrix} \mathbf{S}_{\hat{\boldsymbol{v}}_{11}}(\omega) & \mathbf{S}_{\hat{\boldsymbol{v}}_{12}}(\omega) & \cdots & \mathbf{S}_{\hat{\boldsymbol{v}}_{1,\texttt{NDEGW}}}(\omega) \\ \mathbf{S}_{\hat{\boldsymbol{v}}_{21}}(\omega) & \mathbf{S}_{\hat{\boldsymbol{v}}_{22}}(\omega) & \cdots & \mathbf{S}_{\hat{\boldsymbol{v}}_{2,\texttt{NDEGW}}}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{\hat{\boldsymbol{v}}_{\texttt{NDEGW},1}}(\omega) & \mathbf{S}_{\hat{\boldsymbol{v}}_{\texttt{NDEGW},2}}(\omega) & \cdots & \mathbf{S}_{\hat{\boldsymbol{v}}_{\texttt{NDEGW},\texttt{NDEGW}}}(\omega) \end{pmatrix} \tag{4.32}$$

where, equivalently, all elements can be expressed as a function of $\mathbf{S}_u(\omega)$, $\mathbf{S}_v(\omega)$ and $\mathbf{S}_w(\omega)$.

Some of these large block-matrices in both Equation (4.29) and Equation (4.32) can vanish or be neglected, especially if the correlation between elementary components of the turbulence are neglected (see Section 4.1), that is

$$E\left[\eta_I(t)\mu_I(t+\tau)\right] = 0 \quad \eta, \mu \in \{u, v, w\},\ \eta \neq \mu, \quad \forall I = 1, \ldots, \texttt{NNa} \tag{4.33}$$

Though this is a quite restrictive condition, some more complex model accounting for spatial and time interaction between elementary wind turbulent components could be used [79].

As an example, Figure 4.7-a shows the possible structure of $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau)$ and $\mathbf{S}_{\hat{\boldsymbol{v}}}(\omega)$ for $d = 2$, i.e. $\texttt{NDEGW} = 9$. The diagonal elements represent $\mathbf{S}_u(\omega)$, $\mathbf{S}_v(\omega)$, $\cdots$, $\mathbf{S}_{vw}(\omega)$ at all aerodynamic nodes of the model. The colored boxes indicate where non-zero values can be found. Indeed, cross-correlation functions involving in total odd powers of Gaussian processes vanish. Among all these possible non-zeros values, some might be neglected, especially in these two cases: (i) assuming the wind turbulence components are uncorrelated would discard all elements but the first six blocks of the main block-diagonal (i.e. assuming validity of Equation (4.33)), (ii) assuming a small wind turbulence intensity (i.e. assuming the linearised version of wind forces) would result in neglecting all elements except the first three by three block (upper-left) ones.

Since not all wind components in Equation (4.27) (and so in Equation (4.25) and Equation (4.26)) are Gaussian (as $u^2(t)$ for example), higher statistical moments are required for a full probabilistic description of the buffeting loading. In particular, as soon as quadratic terms are considered ($d \geq 2$), the third rank statistical quantities (i.e. the bicorrelation and the bispectra) need to be established. Thanks to the block matrix representation, the bicorrelation of wind components $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau_1, \tau_2)$ takes the form of a 3-D block-matrix composed of block-tensors $\mathbf{R}_{\hat{\boldsymbol{v}}_{abc}}(\tau_1, \tau_2)$. An equivalent representation at third order consists in the bispectra of the loading, which is formally obtained by the two-fold Fourier transform of the bicorrelation function (see Equation (2.42)). It is noted $\mathbf{B}_{\hat{\boldsymbol{v}}}(\omega_1, \omega_2)$ and is composed of the block-tensors $\mathbf{B}_{\hat{\boldsymbol{v}}_{abc}}(\omega_1, \omega_2)$ corresponding to the bispectra of a given triplet of components a, b, c of wind loading components.

The format of $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau_1, \tau_2)$ and $\mathbf{B}_{\hat{\boldsymbol{v}}}(\omega_1, \omega_2)$ is depicted in Figure 4.7-(b,c), for

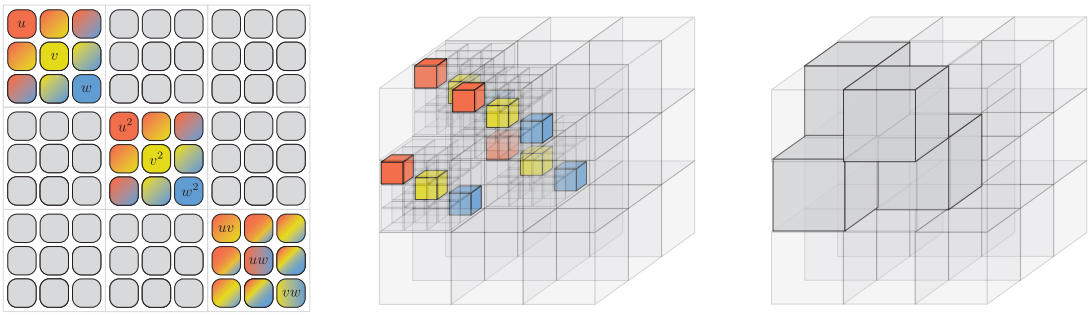


Figure 4.7: (a) Block matrix representation of $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau)$ and $\mathbf{S}_{\hat{\boldsymbol{v}}}(\omega)$; (b-c) Block matrix representation of $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau_1, \tau_2)$ and $\mathbf{B}_{\hat{\boldsymbol{v}}}(\omega_1, \omega_2)$. Quadratic model ($d = 2$). Colors (red, yellow, blue) refer to elementary components u, v, w respectively.

$d = 2$, where the larger shadowed blocks indicate the possible non-zero elements. It is interesting to notice that the first large block in Figure 4.7-c, that is the only remaining block for $d = 1$, is exactly equal to zero. This translates the fact that the loading is Gaussian in case of linearised model. Then, if the cross-correlation between wind turbulence components $u(t)$, $v(t)$ and $w(t)$ is neglected (see Equation (4.33)), in the four large blocks with possible non-zero components, the only ones that remain are the block-tensors colored in red, yellow and blue in Figure 4.7-b.

Higher order statistics could also be developed following the same approach. However, in the following, results will be limited to third order analysis only.

It is emphasised at this stage already that, as soon as $d \geq 2$, the size of matrices $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau)$, $\mathbf{S}_{\hat{\boldsymbol{v}}}(\omega)$ and tensors $\mathbf{R}_{\hat{\boldsymbol{v}}}(\tau_1, \tau_2)$, $\mathbf{B}_{\hat{\boldsymbol{v}}}(\omega_1, \omega_2)$ is much larger than the size of the cross-correlation matrices (or cross-power spectral densities) which constitute the minimum independent information for this problem. To construct these full matrices before projecting them in a modal basis, as seen next, can therefore already be seen as an inefficient approach.

Once the set of wind components (see Equation (4.25) or Equation (4.26)) are characterised in a stochastic manner (i.e. having defined $\mathbf{B}_{\boldsymbol{v}}(\omega_1, \omega_2)$), then also the resulting forces will be easily characterisable as well, since the relation between them is linear (see steps II-III in Figure 4.6). That is, once $\mathbf{B}_{\boldsymbol{v}}(\omega_1, \omega_2)$) is defined, the determination of the `NDOFs` $\times$ `NDOFs` $\times$ `NDOFs` tensor of bispectra of buffeting loads $\mathbf{B}_{\mathbf{f}}(\omega_1, \omega_2)$ is straightforward. For conciseness, each element of such tensor will be denoted as $B_{f_{ijk}}(\omega_1, \omega_2)$, denoting the (ijk)-th element of the 3-D tensor, representing the bispectrum of the buffeting load evaluated at the three i, j and k structural degrees of freedom, i.e. evaluating the bispectrum at the i, j and k-th nodal degree-of-freedom of the I, J and K-th structural nodes

(NNs) respectively. Appendix C provides a detailed mathematical formula.

4.7 Wind model: a specialised case

In Section 4.5.2, a new general nonlinear approach of definition of quasi-steady wind model has been formulated, and the general framework of the nonlinear wind loading model has been presented.

It is now assumed that the block-matrix representation of the PSDs and bispectra of wind velocities are known (see Figure 4.7). In the following developments, only second and third orders will be treated so that these two pieces of information are sufficient. Also, a quadratic loading model is considered ($d = 2$). The small turbulence intensity approximation is formulated and turbulence components are supposed to be uncorrelated so that : (i) only the first three blocks of the main block-diagonal in $\mathbf{R}_{\boldsymbol{v}}(\Delta t)$ and $\mathbf{S}_{\boldsymbol{v}}(\omega)$ remain, and (ii) only the colored blocks in Figure 4.7-b are kept. To assume uncorrelated wind components is a simplification made for this particular example of wind loading; this prevents from having to choose among several existing cross-PSDs of turbulence components, but also to avoid the modeling of the spatial cross-coherence between various components of the wind turbulence. In any case, this simple choice is sufficient to illustrate the concepts developed in this paper. Interested readers could implement the same developments for the general nonlinear wind loading model introduced in Section 4.5.

Under these assumptions, the developments of previous section to a quadratic wind loading model are specialised. The discussion is organised with respect to the modal wind loads, as introduced in Section 3.3. The wind forces at all DOFs of a structural model, $\mathbf{f}(t)$, is given by Equation (4.22), so that the modal force in mode m reads $p_m(t) = \boldsymbol{\phi}_m^T \mathbf{f}(t)$ where $\boldsymbol{\phi}_m^T$ is the m^{th} mode shape. Alternatively, it is also written

$$p_m(t) = \sum_{I=1}^{\texttt{NNa}} \sum_{a=1}^{\texttt{NDEGW}} \alpha_{Ia}^{(m)} \upsilon_{Ia}(t) \tag{4.34}$$

where

$$\alpha_{Ia}^{(m)} = \boldsymbol{\phi}_m^T \mathbf{a}_{Ia} \tag{4.35}$$

is a modal loading coefficient combining the influence of the wind loads in mode m resulting from loads at all degrees-of-freedom resulting from wind load at aerodynamic node I. This formulation takes the same linear combination as for wind load at physical degrees-of-freedom, see Equation (4.20).

In vectorial form (i.e. considering Equation (4.23)) it reads:

$$p_m(t) = \sum_{I=1}^{\texttt{NNa}} \mathcal{A}_I^{(m)} \boldsymbol{\upsilon}_I(t) \tag{4.36}$$

where

$$\mathcal{A}_I^{(m)} = \boldsymbol{\phi}_m^T \mathbf{A}_I = \boldsymbol{\phi}_m^T \mathbf{x}_I \mathbf{a}_I^T = \psi_I^{(m)} \mathbf{a}_I^T \tag{4.37}$$

is a $1 \times \texttt{NDEGW}$ vector grouping the set of modal loading coefficients in Equation (4.35) for a given aerodynamic node I, $\mathbf{A}_I$ is given in Equation (4.24).

4.7.1 Second order

At second order, statistical information is given either by the Autocorrelation function in the time domain, either by the Power Spectral Density (PSD) function in the frequency domain (see Section 3.2.2). In time domain, the cross-correlation function of modal forces in modes m and n is obtained by considering Equation (4.34), which gives

$$R_{p_{mn}}(\Delta t) = \mathbb{E}\left[p_m(t) p_n(t+\Delta t)\right] = \sum_{a=1}^{\texttt{NDEGW}} \sum_{b=1}^{\texttt{NDEGW}} R_{p_{mn}}^{(ab)}(\Delta t) \tag{4.38}$$

where $R_{p_{mn}}^{(ab)}(\Delta t) = \sum_{I=1}^{\texttt{NNa}} \sum_{J=1}^{\texttt{NNa}} \alpha_{Ia}^{(m)} \alpha_{Jb}^{(n)} \mathbb{E}\left[\upsilon_{Ia}(t) \upsilon_{Jb}(t+\Delta t)\right]$ quantifies how the cross-correlation between wind components a and b contributes to the total amount of correlation between the two considered modal forces. Taking the Fourier Transform of Equation (4.38):

$$S_{p_{mn}}(\omega) = \sum_{a=1}^{\texttt{NDEGW}} \sum_{b=1}^{\texttt{NDEGW}} S_{p_{mn}}^{(ab)}(\omega) \tag{4.39}$$

in which $S_{p_{mn}}^{(ab)}(\omega) = \sum_{I=1}^{\texttt{NNa}} \sum_{J=1}^{\texttt{NNa}} \alpha_{Ia}^{(m)} \alpha_{Jb}^{(n)} S_{\upsilon_{Ia}\upsilon_{Jb}}(\omega)$ represents the contribution of the cross-PSD between wind components a and b to the cross-PSD of wind loads in modes m and n.

Under the assumptions considered in this Section, the only components of the wind loading that are necessary are $(a,b) \in \{(1,1),(2,2),(3,3)\}$, so that, in a general fashion,

$$S_{p_{mn}}^{(ab)}(\omega) = \sum_{I=1}^{\texttt{NNa}} \sum_{J=1}^{\texttt{NNa}} \alpha_{Ia}^{(m)} \alpha_{Jb}^{(n)} S_{\upsilon_{Ia}\upsilon_{Jb}}(\omega) \tag{4.40}$$

In vectorial form:

$$S_{p_{mn}}^{(ab)}(\omega) = {\boldsymbol{\alpha}_a^{(m)}}^T \mathbf{S}_{\boldsymbol{\upsilon}_a \boldsymbol{\upsilon}_b}(\omega) \boldsymbol{\alpha}_b^{(n)} \tag{4.41}$$

where

$$\mathbf{S}_{\boldsymbol{\upsilon}_a \boldsymbol{\upsilon}_b}(\omega) = \boldsymbol{\Gamma}(\omega) S_{\boldsymbol{\upsilon}_a} {S_{\boldsymbol{\upsilon}_b}}^T \tag{4.42}$$

is the $\texttt{NNa} \times \texttt{NNa}$ matrix of nodal cross-PSDs for wind components a and b, $\boldsymbol{\Gamma}(\omega)$ is the $\texttt{NNa} \times \texttt{NNa}$ relative nodal spatial coherence function matrix (see Equation

(4.5) and related discussion), $S_{\boldsymbol{v}_a}$ the $\texttt{NNa} \times 1$ vector of PSDs for wind component a evaluated at all NNa nodes.

For example, considering $a = b = 1$, $\upsilon_{Ia} = u_I$ and $\upsilon_{Jb} = u_J$ respectively correspond to the longitudinal component of turbulence at nodes I and J, so that $\mathbb{E}\left[\upsilon_{Ia}(t)\upsilon_{Jb}(t+\Delta t)\right] = R_{u_I u_J}(\Delta t)$, but also

$$S^{(11)}_{p_{mn}}(\omega) = \sum_{I=1}^{\texttt{NNa}} \sum_{J=1}^{\texttt{NNa}} \alpha^{(m)}_{I1} \alpha^{(n)}_{J1} S_{u_I u_J}(\omega) \tag{4.43}$$

where $S_{u_I u_J}(\omega)$ is the cross-PSD of the longitudinal turbulent component $u(t)$ at nodes I and J (see Equation (4.5))

Same developments apply for $a = b = 2$ and $a = b = 3$, noticing however that the coherence function might be adapted for other elementary turbulence components.

4.7.2 Third order

At third order, the bicorrelation function of modal forces m, n and o reads

$$R_{p_{mno}}(\tau_1, \tau_2) = \mathbb{E}\left[p_m(t) p_n(t+\tau_1) p_o(t+\tau_2)\right]. \tag{4.44}$$

Similarly as in Equation (4.38),

$$R_{p_{mno}}(\tau_1, \tau_2) = \sum_{a=1}^{\texttt{NDEGW}} \sum_{b=1}^{\texttt{NDEGW}} \sum_{c=1}^{\texttt{NDEGW}} R^{(abc)}_{p_{mno}}(\tau_1, \tau_2) \tag{4.45}$$

where

$$\begin{aligned} R^{(abc)}_{p_{mno}}(\tau_1, \tau_2) &= \sum_{I=1}^{\texttt{NN}} \sum_{J=1}^{\texttt{NN}} \sum_{K=1}^{\texttt{NN}} \alpha^{(m)}_{Ia} \alpha^{(n)}_{Jb} \alpha^{(o)}_{Kc} \mathbb{E}\left[\upsilon_{Ia}(t) \upsilon_{Jb}(t+\tau_1) \upsilon_{Kc}(t+\tau_2)\right] \\ &= \sum_{I=1}^{\texttt{NN}} \sum_{J=1}^{\texttt{NN}} \sum_{K=1}^{\texttt{NN}} \alpha^{(m)}_{Ia} \alpha^{(n)}_{Jb} \alpha^{(o)}_{Kc} R^{(abc)}_{\upsilon_{IJK}}(\tau_1, \tau_2) \end{aligned} \tag{4.46}$$

represents the contribution of the bicorrelation function of elementary wind components a, b and c to the bicorrelation of wind loads in modes m, n and o. Application of the double Fourier Transform to Equation (4.45) gives the bispectrum of modal loads

$$B_{p_{mno}}(\omega_1, \omega_2) = \sum_{a=1}^{\texttt{NDEGW}} \sum_{b=1}^{\texttt{NDEGW}} \sum_{c=1}^{\texttt{NDEGW}} B^{(abc)}_{p_{mno}}(\omega_1, \omega_2) \tag{4.47}$$

where $B_{p_{mno}}^{(abc)}(\omega_1, \omega_2)$ represents the contribution of triplet (a, b, c) to the total bispectrum. This contribution can be expressed as a function of the power spectral density of the turbulence components as discussed earlier. Under the assumptions made in this Section, only some triplets (a, b, c) do contribute to the bispectrum of wind loads, which are the 4 square boxes in a more dark grey colour in Figure 4.7-c

Example

For instance, for $a = 4$ and $b = c = 1$ (component involving $u_I^2(t)$, $u_J(t)$ and $u_K(t)$ in Equation (4.27)), the contribution $R_{p_{mno}}^{(abc)}(\tau_1, \tau_2)$ to the autocorrelation becomes

$$R_{p_{mno}}^{(411)}(\tau_1, \tau_2) = \sum_{I=1}^{\mathtt{NN}} \sum_{J=1}^{\mathtt{NN}} \sum_{K=1}^{\mathtt{NN}} \alpha_{I4}^{(m)} \alpha_{J1}^{(n)} \alpha_{K1}^{(o)} \mathbb{E}\left[u_I^2(t) u_J(t+\tau_1) u_K(t+\tau_2)\right]. \quad (4.48)$$

It can be proved that [5], after developing the expectation operator, Equation (4.48) becomes

$$R_{p_{mno}}^{(411)}(\tau_1, \tau_2) = \sum_{I=1}^{\mathtt{NN}} \sum_{J=1}^{\mathtt{NN}} \sum_{K=1}^{\mathtt{NN}} \alpha_{I4}^{(m)} \alpha_{J1}^{(n)} \alpha_{K1}^{(o)} \left(R_{u_I u_J}(\tau_1) R_{u_I u_K}(\tau_2) + \sigma_u^2 R_{u_J u_K}(\tau_2 - \tau_1)\right) \quad (4.49)$$

In the frequency domain, the important contribution of the triplet (u^2, u, u) to the total cross-bispectrum of the modal force (m, n, o) is

$$B_{p_{mno}}^{(411)}(\omega_1, \omega_2) = \sum_{I=1}^{\mathtt{NN}} \sum_{J=1}^{\mathtt{NN}} \sum_{K=1}^{\mathtt{NN}} \alpha_{I4}^{(m)} \alpha_{J1}^{(n)} \alpha_{K1}^{(o)} S_{u_I u_J}(\omega_1) S_{u_I u_K}(\omega_2) \quad (4.50)$$

since the term depending on the single autocorrelation function $R_{u_J u_K}(\tau_2 - \tau_1)$ does not contribute to the cumulant (it actually provides a Dirac delta function in the frequency domain).

Other triplets, expressed as similar combinations of $u_I^2(t)$, $u_J(t)$ and $u_K(t)$ (one squared and two original values) can be derived in the same way. In the following, the mixed contributions coming from mixed combinations of $u(t)$, $v(t)$ and $w(t)$ at different nodes are disregarded. This is direct consequence of neglecting any possible spatial (and temporal) coherence that might exist between the elementary wind turbulence components [51, 52]. Similar formulations can be retrieved for all the triplets (a, b, c) of wind components contributing, under the considered assumptions, to the construction of the full 3D matrix of bispectra of modal forces. They are not shown in detail. However, they are obtained with a similar approach as shown above for deriving Equation (4.50).

4.8 Proper Orthogonal Decomposition in Wind Engineering

Wind Engineering, as it is known today, is a fairly young discipline born around 1950. First pioneering work in this matter is attributed to Alan G. Davenport [28, 50, 80]. While still highly influenced by other disciplines such as aerospace, climatology and meteorology, the main focus was (and still is) the quantification of the wind action on slender structures (e.g. skyscrapers, long bridges, towers, etc.). The configuration of these new kind of structural systems have shown to be highly susceptible to the wind, so that the need of new engineering practices arose to account for such wind-structure interactions. Not only under a safety point of view, but also in terms of serviceability of structures with respect to human comfort [51]

Application of POD techniques in Wind Engineering related fields has been quite extensively resumed and discussed by Solari et. al in [43]. A brief historical and mathematical overview of the different POD variants that have been formulated over the last decades can be found in Appendix B.

In following sections, first a state of the art of use of POD techniques in wind engineering practices is recalled. This will serve as an accompanying mean to the novel formulation of the extension of POD techniques to non-Gaussian buffeting analyses. Also, any mathematical development will be based on the Spectral Proper Orthogonal Decomposition (SPOD) formulation. For details of its mathematical formalism, see Appendix B.

4.8.1 Mathematical background

This section will shortly recall the mathematical developments involved in applications of POD to wind turbulence fields, and so to the solution of the general dynamic problem discussed in Section 3.3, at second statistical order. This is extensively discussed in [81] and [82].

It will also serve to extend its formulation to bispectral analyses.

Under a very general point of view, $S_{u_I u_J}(\omega)$ might be seen as a single element of the (possibly large) Cross Power Spectral Density Matrix (CPSDM) [81] of wind turbulence at the (aerodynamic) nodes of the model (see Equation (4.5)). As an example, the CPSDM for the longitudinal component $u(t)$ is commonly defined as $\mathbf{S}_u(\omega)$. The spectral decomposition of $\mathbf{S}_u(\omega)$, obtained by solving

$$[\mathbf{S}_u(\omega) - \lambda_p(\omega)\mathbf{I}]\,\boldsymbol{\psi}_p(\omega) = \mathbf{0} \qquad (p = 1, \ldots, \mathtt{NN}) \tag{4.51}$$

reads

$$\mathbf{S}_u(\omega) = \boldsymbol{\Psi}(\omega)\ \boldsymbol{\Lambda}(\omega)\ \boldsymbol{\Psi}^T(\omega) \tag{4.52}$$

where $\mathbf{\Lambda}(\omega)$ and $\mathbf{\Psi}(\omega)$ are the (frequency dependent) diagonal eigenvalues and eigenvector matrices respectively, resulting from the eigenproblem in Equation (4.51) (see Appendix B). It is assumed that the eigenvalues are sorted in descending order in $\mathbf{\Lambda}(\omega)$. All of them are positive because of the positivity of the CPSDM $\mathbf{S}_u(\omega)$. Keeping only the first $\mathtt{N_p} \ll \mathtt{NN}$ modes with significant eigenvalues, Equation (4.52) can be rewritten as

$$S_{u_I u_J}(\omega) = \sum_{p=1}^{\mathtt{N_p}} \psi_{Ip}(\omega)\lambda_p(\omega)\psi_{Jp}(\omega). \tag{4.53}$$

Injecting Equation (4.53) in Equation (4.40):

$$S^{(11)}_{p_{mn}}(\omega) = \sum_{p=1}^{\mathtt{N_p}} \sum_{I=1}^{\mathtt{NN}} \sum_{J=1}^{\mathtt{NN}} \alpha^{(m)}_{I1}\alpha^{(n)}_{J1}\psi_{Ip}(\omega)\lambda_p(\omega)\psi_{Jp}(\omega) \tag{4.54}$$

which could seem to be more involved but, swapping summations and writing more generally, it becomes

$$S^{(ab)}_{p_{mn}}(\omega) = \sum_{p=1}^{\mathtt{N_p}} \zeta^{(m)}_{pa}(\omega)\lambda_p(\omega)\zeta^{(n)}_{pb}(\omega) \tag{4.55}$$

where

$$\zeta^{(m)}_{pa}(\omega) = \sum_{I=1}^{\mathtt{NN}} \alpha^{(m)}_{Ia}\psi_{Ip}(\omega). \tag{4.56}$$

The double sum over the number $\mathtt{NN}$ of loaded (aerodynamic) nodes as in Equation (4.40) has dropped in place of a single sum over the $\mathtt{N_p} \ll \mathtt{NN}$ kept modes of the spectral decomposition of $u(t)$.
Therefore, in general, a total of

$$\mathtt{N_p} \cdot (\mathtt{NN} \cdot \mathtt{NDEGW} \cdot \mathtt{NM})$$

operations are needed to compute an element of the $\mathtt{NM} \times \mathtt{NM}$ matrix of PSDs of modal forces $S_{p_{mn}}(\omega)$, compared to the

$$(\mathtt{NN} \times \mathtt{NDEGW})^2$$

of the classic approach as of Equation (4.40). It just requires the prior computation and storage of $\zeta^{(m)}_{pa}(\omega)$ for $a \in \{1, \ldots, \mathtt{NDEGW}\}$, and $m \in \{1, \ldots, \mathtt{NM}\}$. Assuming a quadratic wind model transformation (see Section 4.7), then $\mathtt{NDEGW} = 10$ (9 if the constant term is not considered), and generally, for large structures,

$\mathtt{NM} \approx \frac{\mathtt{NN}}{100}$ or sometimes even less, so that in total

$$\mathtt{N_p} \cdot \left(\mathtt{NN} \cdot 10 \cdot \frac{\mathtt{NN}}{100} \right) = \mathtt{N_p} \cdot \left(\frac{\mathtt{NN}^2}{10} \right)$$

which compared with the original number of operations, results in a saving factor of around

$$\frac{10 \cdot \mathtt{NDEGW}^2}{\mathtt{N_p}} = \frac{1000}{\mathtt{N_p}}.$$

Hence Equation (4.55) is definitely much more efficient to implement than Equation (4.40), specially for those cases where using just a few wind modes $\mathtt{N_p}$ is enough to capture a good amount of energy of the original wind turbulent field, say 90% of the total energy.

Then, by including common Stochastic Analysis introduced in Sections 3.2.3 and 3.3, the CPSDM of modal responses is computed as

$$S_{q_{mn}}(\omega) = H^{\star}_{mm}(\omega) \left(\sum_{a=1}^{\mathtt{NDEGW}} \sum_{b=1}^{\mathtt{NDEGW}} \sum_{p=1}^{\mathtt{N_p}} \zeta^{(m)}_{pa}(\omega) \lambda_p(\omega) \zeta^{(n)}_{pb}(\omega) \right) \overline{H^{\star}_{nn}}(\omega) \tag{4.57}$$

or in vectorial form

$$S_{\mathbf{Q}}(\omega) = \sum_{p=1}^{\mathtt{N_p}} \mathbf{H}^{\star}(\omega) \mathbf{Z}_p(\omega) \lambda_p(\omega) \overline{\mathbf{Z}_p}(\omega) \overline{\mathbf{H}^{\star}}(\omega) \tag{4.58}$$

where $\mathbf{Z}_p(\omega)$ is a $\mathtt{NM} \times \mathtt{NDEGW}$ matrix grouping coefficients $\zeta^{(m)}_{pa}(\omega)$ for all modes m and wind coefficients a.

The CPSDM of the resulting structural responses $\mathbf{x}(t)$ then reads:

$$S_{\mathbf{x}}(\omega) = \sum_{p=1}^{\mathtt{N_p}} \mathbf{\Phi} \mathbf{H}^{\star}(\omega) \mathbf{Z}_p(\omega) \lambda_p(\omega) \overline{\mathbf{Z}_p}(\omega) \overline{\mathbf{H}^{\star}}(\omega) \mathbf{\Phi}^T. \tag{4.59}$$

Equation (4.59) formulates the relation (in a stochastic manner) between the structural responses (the outcome of the equations of motion of the dynamic problem) and the POD decomposition of the base input, the spatial wind turbulent field $(u(t), v(t), w(t))$.

4.8.2 Extension of POD formulation to bispectral analyses

The assumption of Gaussian wind buffeting loading has been extensively used across all practitioners in the wind engineering field. The reasons why this has

been the daily practice have been discussed in previous sections (see Section 4.5), which to recall, stand in the fact the resulting wind loads are Gaussian if turbulence is assumed to be a Gaussian process. As a consequence, in cases of linear systems, the structural responses will be Gaussian too, for which only the first two statistical moments (mean and variance) are needed for a complete probabilistic description (see Section 2.2). Indeed, many are the cases where this simplification is admitted by the physical configuration of the studied case. However, not always this is a safe assumption to make, neglecting a priori the intrinsic non-Gaussian nature of the wind buffeting loading (see Section 4.5.2).

It has also been mentioned that some authors have already done works in which they did not neglected the non-Gaussian nature of wind loads, and comparing results with a Gaussian analysis [8, 12]. However, these applications were limited to small systems, with less than a dozen degrees-of-freedom.

If the non-Gaussian nature of wind loads is considered and the PSD matrix and the tensor of bispectra of modal wind loads are determined, the analysis can follow the general steps described in Section 3.3, up to third statistical order.

Clearly, with today's advancements in technological facilities, construction and evaluation of the full matrix of cross-PSDs of nodal wind loads (see Equation (4.42)) would not be a problem, even for large structural systems with more than thousands degrees-of-freedom.

However, it is computationally demanding to construct such a huge amount of information, based on the single PSD matrices of wind turbulence components.

The applied concepts are the same as in Equation (4.54), but their application to third order statistics shows some substantial differences.

Following the concepts explained in detail for Eqs. (4.52) and (4.53), it is possible to express the two PSDs of wind turbulence as a sum of some principal components

$$\begin{aligned} S_{u_I u_J}(\omega_1) &= \sum_{p=1}^{\mathtt{N_p}} \psi_{Ip}(\omega_1)\lambda_p(\omega_1)\psi_{Jp}(\omega_1) \\ S_{u_I u_K}(\omega_2) &= \sum_{q=1}^{\mathtt{N_q}} \psi_{Iq}(\omega_2)\lambda_q(\omega_2)\psi_{Kq}(\omega_2). \end{aligned} \tag{4.60}$$

Injecting Equation (4.60) into Equation (4.50) yields

$$B^{(411)}_{p_{mno}}(\omega_1,\omega_2) = \sum_{p=1}^{\mathtt{N_p}}\sum_{q=1}^{\mathtt{N_q}} \zeta^{(n)}_{p1}(\omega_1)\,\zeta^{(o)}_{q1}(\omega_2)\,\chi^{(m)}_{pq4}(\omega_1,\omega_2)\,\lambda_p(\omega_1)\lambda_q(\omega_2) \tag{4.61}$$

where

$$\chi^{(m)}_{pqa}(\omega_1,\omega_2) = \sum_{I=1}^{\mathtt{NN}} \alpha^{(m)}_{Ia}\psi_{Ip}(\omega_1)\,\psi_{Iq}(\omega_2)\,. \tag{4.62}$$

The most significant difference with respect to POD applied in a Spectral analysis is the dependence to not just one but two frequencies, ω_1 and ω_2, as a direct consequence of the definition of the bispectrum (see Equation (2.42)). Moreover, the term $\chi_{pqa}^{(m)}(\omega_1, \omega_2)$ detailed in Equation (4.62) and appearing in Equation (4.61) couples the dependence on the two independent variables ω_1 and ω_2, which prevents any mathematical simplification, for example by splitting the dependence from the two independent variables ω_1 and ω_2.

There is no doubt that the double summation over loading modes in Equation (4.61) is much more efficient than the three-fold summation over the total number of aerodynamic nodes in Equation (4.50). The computational cost to estimate each $\zeta_{pa}^{(m)}(\omega)$ requires a loop on all aerodynamic nodes, and this operation needs to be done for combinations (m, p, a), i.e. $\mathtt{NM} \times \mathtt{N_p} \times \mathtt{NDEGW}$. The computational cost to estimate each $\chi_{pqa}^{(m)}(\omega_1, \omega_2)$ also requires a loop on all nodes, and this operation needs to be done for combinations (m, p, q, a), i.e. $\mathtt{NM} \times \mathtt{N_p} \times \mathtt{N_q} \times \mathtt{NDEGW}$. As a consequence, the computational intensity required to compute an element of the 3D matrix of modal forces bispectra $B_{p_{mno}}(\omega_1, \omega_2)$ amounts to $\mathtt{NN}^3 \times \mathtt{NDEGW}^3$ in a normal approach as of Equation (4.50), and to $\mathtt{N_p} \times \mathtt{N_q} \times \mathtt{NN} \times \mathtt{NDEGW} \times \mathtt{NM}$ in case of application of POD techniques. This leads to a potential iterations ratio of

$$\frac{(\mathtt{NN} \times \mathtt{NDEGW})^3}{\mathtt{N_p} \times \mathtt{N_q} \times \mathtt{NN} \times \mathtt{NDEGW} \times \mathtt{NM}} = \frac{(\mathtt{NN} \times \mathtt{NDEGW})^2}{\mathtt{N_p} \times \mathtt{N_q} \times \mathtt{NM}} = \frac{\mathtt{NN} \times 10000}{\mathtt{N_p} \times \mathtt{N_q}}$$

which can reach considerably high values if the required number of modes to be kept $\mathtt{N_p}$ and $\mathtt{N_q}$ are small with respect to the number of (aerodynamic) nodes $\mathtt{NN}$.

Chapter 5

Numerical Analysis & Algorithmic Development

This Chapter is dedicated to the numerical side and algorithmic developments that have been carried out in the context of this Thesis' work.

5.1 Problem specificities

5.1.1 Morphology of the bispectrum of wind load

In the process of developing an algorithmic arrangement, the determination of the specificities of the problem under investigation is indeed the starting point.

In Appendix C, Equation (C.1) details the complete mathematical formulation of the bispectra of the quadratic non-Gaussian wind buffeting forces at the level of aerodynamic nodes. It is a sum of many terms, each dependent on the product of two PSDs of the elementary wind turbulent components $(u(t), v(t), w(t))$, which are assumed to be low frequency Gaussian random processes (see Section 3.2.2). Figure 5.1 reports a typical wind turbulence PSD, obtained from time records. It clearly shows how most of the energy is contained in the lower range of frequencies, while at medium-to-high ranges, the energy content drops quickly.

Since the determination of the bispectrum of non-Gaussian wind forces directly depends on a linear combination of products of turbulence PSDs (see Equation C.1), it is expected that the bispectrum of wind loads shows a similar behaviour, but in a two-dimensional frequency space. As a consequence, if the PSDs of wind turbulence, and so the resulting wind forces, independently to the degree of the translation model, show a major frequency content near the origin $\omega = 0$, the bispectrum shows a its frequency content near the origin $(\omega_1, \omega_2) = (0, 0)$ as well, where most of the energy (i.e. volume in this case) is concentrated, as shown in Figure 5.2-a.

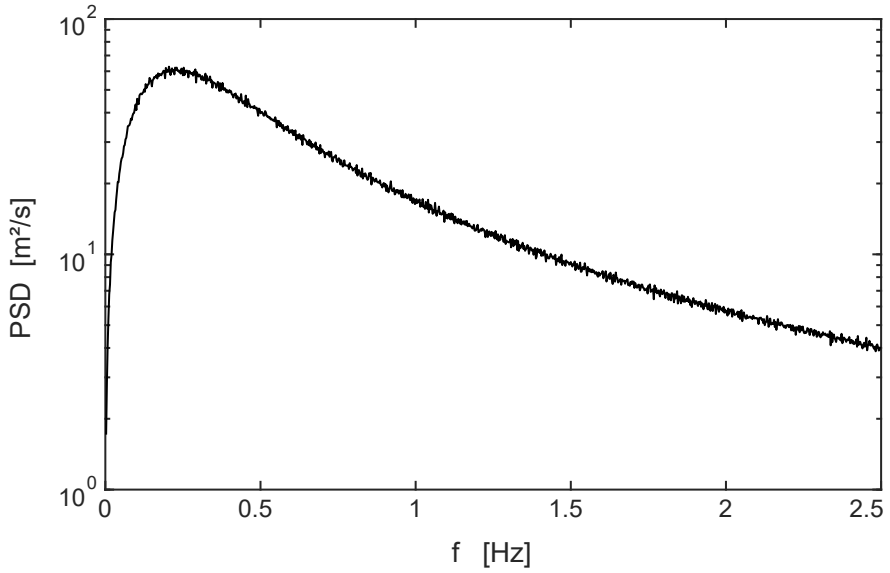


Figure 5.1: Example of a common PSD curve of wind turbulence (log scale).

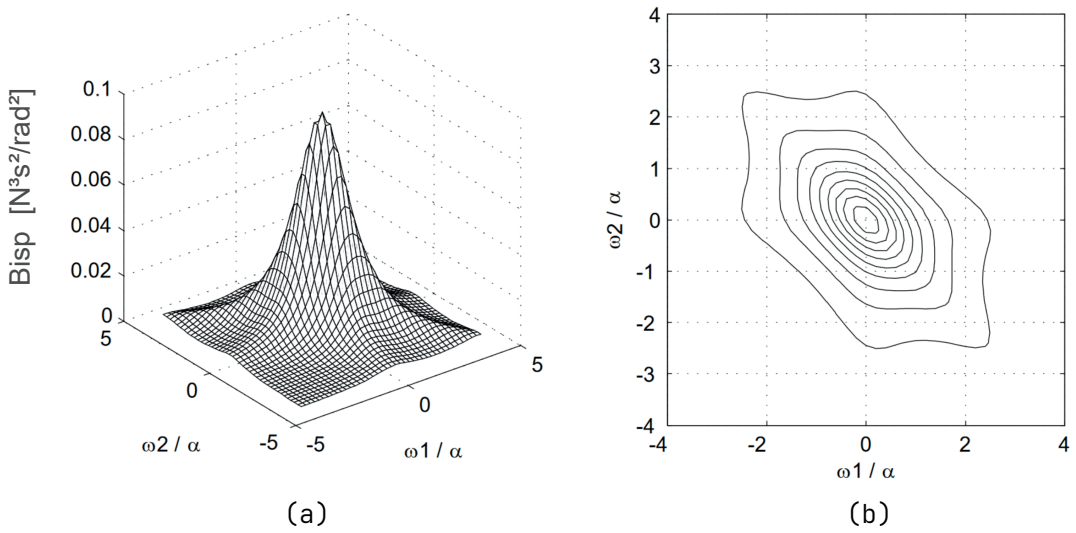


Figure 5.2: Example of common bispectrum of wind loads. (a) 3D view. (b) Above (plane) view. Taken from [5].

Near the origin, the (bi-)spectra of buffeting forces exhibits a considerable gradient. It is crucial, in a discretisation process, to use a right amount of points to correctly capture and represent its sharp changes. On the other hand, as soon as one starts to move far from the origin, the gradient significantly diminishes, up to a point where the decaying is so small that it can be almost assumed constant.

Looking at $B_f(\omega_1, \omega_2)$ illustrated in Figure 5.2-b, one notices that, on plane the shape looks more as an ellipsis rather than a circle, with its strong axis coincident with the bisect of 2^{nd} and 4^{th} quadrants. This is caused by the coupling terms, i.e. the terms in Equation (C.1) that contain wind turbulence PSDs depending either on the sum or on the difference of the two independent variables (ω_1, ω_2).

On the other hand, when close enough to the origin, great care must be employed when discretising this fast varying function. The following sections will further clarify this point, but for the sake of clarity, it must be stressed out that a good representation of the probabilistic quantities of the loading is the most important step when it comes to spectral analyses in general, and even more in context of higher-order analyses such as bispectral analyses.

5.1.2 A problem of fundamental numerical nature

Having the best possible estimate of the statistical moments of the loading is fundamental for a good estimate of the same moments of the resulting structural responses (e.g. displacements, internal stresses, internal efforts, etc.), specially in cases where the overall structural behaviour is dominantly quasi-static. Since this estimate is the result of the numerical integration of some sharp functions in the proximity of the origin, a good choice of discretisation points is the key to achieve the desired precision limiting the computational burden. In fact, in general spectral analyses fall in the category of those problems for which there exists no closed-form solution. The only way to solve them is numerically, either by means of approximations or by discretising its a-priori continuous domain, transforming it into a large enough number of finite portions. Because an exact reference solution may not be found, some techniques have been developed in the past to quantify accuracy of the results (e.g. convergence methods, [83]). Yet, all these numerical techniques exist as a way to possibly quantify and limit the error in the estimate, but which are themselves affected by the same issues. In such complex cases, where the result is of pure numerical nature, one has to accept that the generated result can be at best very close to the theoretical solution, but in most of the cases, it will be just close enough for the result to be considered acceptable, particularly in engineering applications. Indeed, when it comes to measurements, the sources of errors are various. If on one side, one could not control the errors intrinsic to the tools used for the computation itself (e.g. systematic environmental errors), on the other one should limit and possibly avoid any induced gross error, i.e. errors due to wrong computations. The reasons of such errors could be very wide, but in any case they can and must be avoided. It is here were the engineering judgement is required.

It is interesting to note that the integral of a PSD or bipsectrum of wind load is known analytically, for SDOFs at least. This is indeed a very important source

of results verification, which has been extensively used in [5]. Such verifications for SDOF systems will not be treated in this Work specifically, since the main focus is to extend the Bispectral analysis to possibly large MDOF linear systems.

5.1.3 The loading and response scales: the different morphology of the Frequency Response Functions

Bispectra of buffeting loading has a sharp peak around the origin (ω_1, ω_2) of the 2D frequency space, which gradually diminishes until it looks almost flat and small. Equation (3.45) has then introduced the direct mathematical formulation for computing the cross-bispectra of modal forces of the triplet of modes (m, n, o). Basically, it shows that it is obtained as a linear combination of crossed-bispectra of nodal loads, modulated by some linear functions, namely the natural mode shapes of vibration of the system (e.g. structure). Consequently, the (bi-)spectra of modal loads exhibits the same shape as those of nodal loads, simply rescaled. This supposes that dynamic analysis is carried out in a modal basis, as it has been extensively explained and justified in previous sections (see Section 3.1 for instance). In such cases, once the modal loading has been characterised either in a direct manner (time series) or in a stochastic fashion via its statistical moments, the general equations of motion (3.2) have to be solved, either directly in time domain via time-marching algorithms such as Newmark [14], or in the frequency domain by means of the stochastic dynamic equations formulation (see for instance Equation (3.39) for a formulation at 2^{nd} statistical order). In the latter case, the Frequency Response Function $\mathbf{H}^{\star}(\omega)$ detailed in Equation (3.5) is also a function that varies with frequency. It has very sharp peaks in correspondence of the natural frequencies (see middle image in Figure 1.2-b). In fact, the FRF acts as a filter and amplifies the structural response subjected to a given dynamic loading whenever the frequency content of both loading and dynamic response are close enough. In the limit case when the two frequencies match, then the generating phenomenon is called resonance. For bispectral analysis, thus at third statistical order, the concepts are similar. Nonetheless, if at second order the FRF has a single peak, at third order the 3^{rd} order FRF, also called the 2^{nd} order Volterra kernel (see Equation (3.47)) shows six primary resonance peaks in correspondence of points $(0, \pm\omega_i)$, $(\pm\omega_i, 0)$ and $(\pm\omega_i, \mp\omega_i)$. There are also other six secondary peaks located at $(\pm\omega_i, \pm\omega_i)$, $(\pm 2\omega_i, \mp\omega_i)$ and $(\pm\omega_i, \mp 2\omega_i)$.

In case of a MDOF system with $\texttt{NM}$ modes of vibrations, if at 2^{nd} order one would expect at most $\texttt{NM} + 1$ peaks, at 3^{rd} order the bispectra of structural responses can have up to $(\texttt{NM} * 6) + 1$ peaks. However, in reality when it comes to the morphology of cross-bispectra as presented in Section 3.2.2.4, things are even more complex, with the appearance of some other bi- and tri-resonant peaks

(see for instance [33]) alongside simple resonant ones, such as those shown in Figure 1.2-c. Figure 5.3 shows detailed graphical representation of such other peak types, many of them are only found in cross-bispectra.

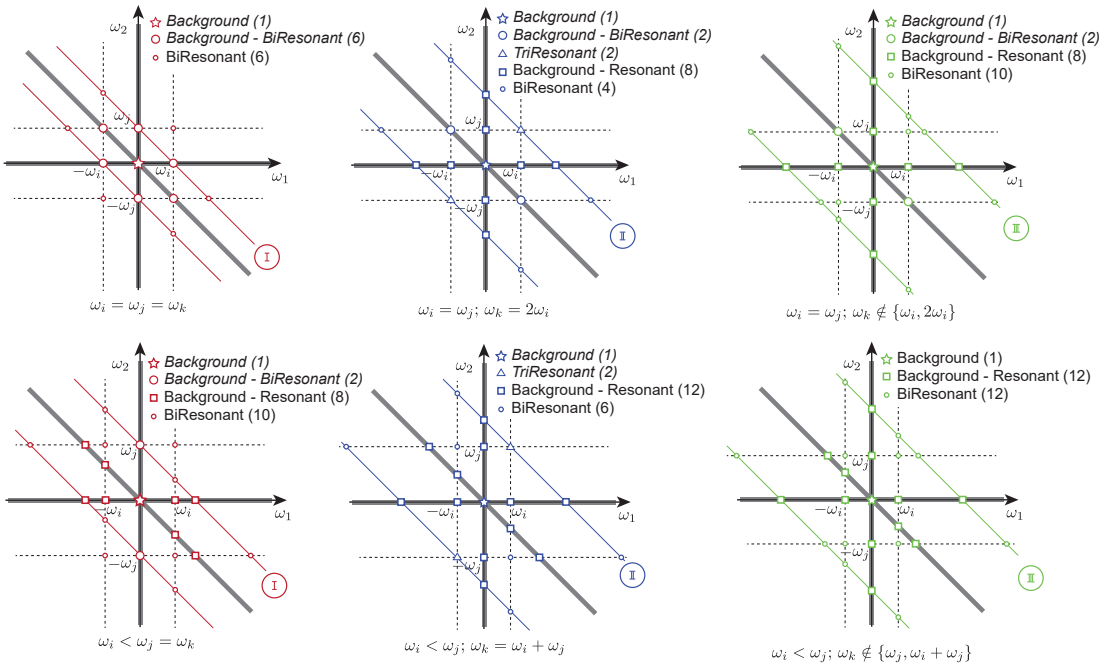


Figure 5.3: Graphical representation of all types of peaks present in a general bispectrum of modal responses in a triplet (i, j, k) of modes, with natural frequencies $\omega_i, \omega_j, \omega_k$.

It should now be easy to understand and visualise the clear difference in shape of spectra, at both second and third statistical orders, between the loading process and the resulting structural response, after having introduced the influence of the corresponding Volterra kernel spectra of each natural vibration mode illustrated in Figure 1.2-c. In a general fashion, the spectra of the loading process contains most of its energy close to the origin, where the kernel function is almost flat and small. On the contrary, where the Volterra kernel function shows its resonant peaks, the spectrum of the buffeting loading shows relatively reduced amplitudes and varies little over the characteristic width of the resonant peaks. The combination of these two aspects result in a kind of cancellation of effects of the applied load to the resulting structural response. On the contrary, for the cases where the frequency contents of the load and that of the structure natural response get close, resonance effects may arise, resulting in an important structural response.

5.1.4 A unique solution that satisfies all cases?

Recalling the considerations made at the beginning of the Chapter, great care has to be adopted when discretising these sharp functions. However, different discretisations are needed for accurately capturing both the spectra of the loading and the natural structural response. The spectra of the buffeting loading needs a dense discretisation close to the origin, while the spectra of the natural structural response require dense sampling close to the resonance peaks, that are close to the natural frequencies of each vibration mode. These two conditions are essential to avoid considerable representation errors. In fact, there are 3 possible scenarios:

1. Using the optimal discretisation of the loading spectra, which would cause considerable errors in the discretisation of the Volterra kernels;

2. Using the optimal discretisation of the Volterra kernels, causing errors in the discretisation of the spectra of the loading;

3. Using the union of the two optimal discretisation patterns to discretise both the spectra of both the loading and the Volterra kernels.

If points 1 and 2 are clearly inefficient because of important loss of information at the level of discretisation of either the loading or the structural Volterra kernel spectra, point 3 is actually good under that point of view, since it would not suffer from any information loss. However, in the context of higher order spectral analyses, such as bispectral analyses, the FRF is characterised by more than a single peak, meaning that there are multiple regions where the gradient is important. They hence require more than a single dense discretisation zone to be correctly represented. On the other hand, the spectra of the buffeting load always have their peaks close to the origin, which is easily discretisable using a single dense discretisation zone centred at the origin. Therefore, applying also the discretisation pattern needed to correctly represent the structural Volterra kernels spectra to the one of the loading process would enforce a considerable overhead, that would significantly slow down computations, without much gain in terms of accuracy. The slow down is mainly due to the numerical complexity in evaluating the bispectral information of the wind load, by applying either Equation (3.45), or Equation (4.61) if the POD formulation to bispectral analyses is considered.

It is clear that the best approach is having the optimal discretisation for each of the spectra of loading and response to be represented to avoid any gross error in their representation, as well as to ensure an appropriate balance between resources and computational time on one side and gain in accuracy and efficiency on the other. The unique features that characterise the Bispectral problem are the same time a positive and negative aspect. Positive because it can allow for specific simplifications, which might not be applicable in the general case. Negative

because at the same time this might impose constraints on how the problem must, or can, be tackled. In that, standard meshing approaches (and related integration) process might have been used, at least in a first stage of developments. There exists several good works on adaptive 2D/3D mesh generation, which are well established on an International level [84, 85, 86]. However, considering the specificities of the Bispectral problem, it has been decided to develop an ad-hoc, optimal discretisation and integration scheme, instead of making use of already existing, powerful meshing tools. The main reason is that they might be overshot to tackle this problem, for which a possibly much simpler and direct solution can suffice without requiring too much mathematical and computational complexity. Discussions will follow to clarify and justify this choice, to prove that this was indeed the best option for this specific problem. This implementation will have strong focus on the two most important macro areas: (i) the background region, close to the origin (i.e. 0 and $(0,0)$), and (ii) all the resonance peaks.

5.2 The novel MESHER algorithmic arrangement

In the following, an accurate but fast enough method is presented, for the determination of bispectra and estimates of the third order statistical moment of non-Gaussian wind loading and resulting non-Gaussian structural responses. This algorithm constitutes the core of this Thesis. Most of the concepts that will be discussed are formalised to 3^{rd}-order analyses, but can be easily downgraded and adapted to second statistical order by simply making the second spatial dimension in the 2D frequency space collapse.

While the concept of discretisation is indeed very broad, in the context of (bi-)spectral analyses it is mainly applied to lines (e.g. PSDs) and surfaces (e.g. bispectra) elements. As quite extensively explained in previous section, an optimal discretisation of both (bi-)spectra of buffeting loading and resulting structural response would require a different optimal pattern. Indeed, once the domain of existence of a given function is well defined, defining sub-parts of it is quite straightforward. For this, the concept of a MESHER has been introduced in the algorithmic arrangement. As the word semantic already suggests, its main goal is that of meshing, i.e. creating a finite discretisation of a given domain. In this specific case, the domain is represented by the 2D frequency space.

The projection of the spectra of loading in the modal basis is definitely the most expensive operation. Therefore, this operation requires the greatest care in minimising the impact on the overall computational cost, while maintaining the desired level on overall precision.

The mathematical side is provided by Section 4.8, which provided the novel formulation of POD decomposition of the turbulent wind field generating non-Gaussian wind forces. This significantly speeds up the computation, since it allows dropping at least one sum operator over the ensemble of the number of aerodynamic nodes `NNa` (see Equation 4.61), compared to the conventional approach.

The role of the next sections is to find a balance between two extreme and unacceptable solutions: (i) very fast but inaccurate or (ii) very accurate but extremely slow computations.

5.2.1 The concept of "zone"

The power of the MESHER approach stands in its elementary *zones*. In such context, a zone represents an independent sub-part of the global domain having a simple geometrical shape (for instance triangular, rectangular or circiular shaped), having its own internal discretisation.

One might certainly ask the reasons behind the introduction of such concept of zone. As per its conception and definition, each zone's independence allows for the maximum flexibility in terms of their internal discretisation, once each zone has been uniquely placed within the global domain. In other words, once a zone has been uniquely defined, it will have its own unique policies of internal definition and management, completely unaware of the possible existence of other zones in the surroundings. Such zones could be theoretically placed anywhere within the limits of the global domain, being it the vector (e.g. spectral, 1D) or space (e.g. bispectral, 2D) of frequencies. Nonetheless, at the time of their definition, the condition of non-overlapping should be ensured, to avoid (i) possible duplicate points and (ii) consequent overestimations (or underestimations) in resulting statistical moments, which are the result of the integration of such complex functions (i.e spectra and bispectra) evaluated at these points. This aspect will be further discussed and explained in the next section. All in all, this allows for great flexibility, specially useful in cases where within the global domain some areas might be of higher importance than others, as for the peak areas in a spectra or bispectra. The zones that will end up patching those important areas will consequently have stricter internal policies with respect to those located far from the peaks. Clearly, this would be hard to achieve with such a degree-of-freedom other than splitting the global domain in some independent sub-spaces (i.e. zones). Otherwise, one would end up in the situation where a stricter condition of existence would be inevitably enforced to all other sub-parts where in reality much lighter conditions would have already met the required local accuracy criterion.

However, some caution has to be employed in order to fully unlock its potential. In fact, an important detail resides in the shapes that such elementary zones are allowed to take. All in all, allowing the definition of zones with complex shapes would automatically invalidate much of the reasoning that have brought to the development of the MESHER algorithm. In fact, it is recalled that the most important information when it comes to a bispectral analysis, is an accurate representation of the bispectrum of the loading. This is crucial for ensuring accuracy at the level of bispectra, and so third order statistical moments, of structural responses. A wrong choice of the basic shapes that these zones can take could lead to an enormous amount of information (i.e. bispectra of the loading), together with an algorithmic complexity, that would need to be stored to be able to compute bispectra of structural responses. Possibly, allowing the use of simple shapes for each zone could lead to a considerably less amount of information to be stored, as well as a much simpler algorithmic complexity.

In this context, such basic shapes are chosen to be two in the MESHER approach: (i) triangles and (ii) rectangles. In that, it is know from other scientific domains such as computer graphics (where spatial continuum discretisation is at the core) that lines (2D) and triangles (3D) are the most basic geometrical entities with which every other possible complex shape can be split into, and represented with [87, 88, 89]. However, in the context of the MESHER algorithm, rectangles are also considered as an independent basic zone shape, even though they are representable by means of the union of two triangles. This has been chosen on purpose since avoiding such 2-triangles representation of a rectangle would considerably ease definition as well as treatment of such rectangular zones. More explanations will follow in Section 5.2.1.2.

5.2.1.1 The triangular zone

A triangle is the most simple entity in which all kinds of (spatial) geometries can be broken down into, when it comes to discretisation. Figure 5.4-a shows the most generic way to represent and uniquely identify a triangle in a 2D planar space.

Among all possible ways, a triangle could be uniquely defined[1] by knowing:

- The three sides length;
- Two sides length and the angle in between; the third side is then determined by means of the cosine rule;
- One side length and two angles;
- Coordinates of all three vertices.

[1] Assumed the conditions of existence are met.

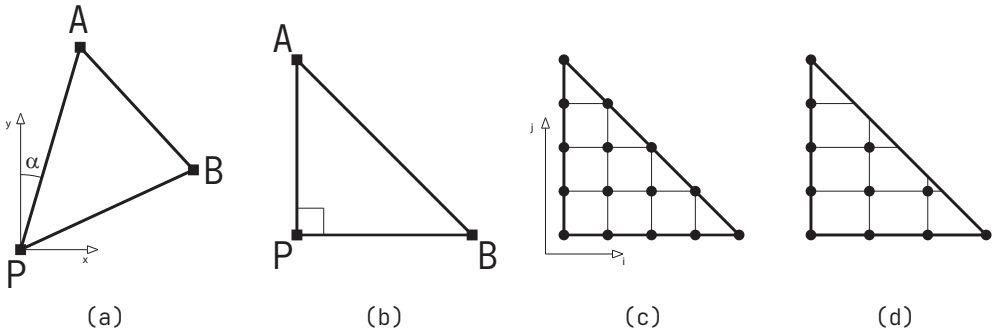


Figure 5.4: (a) Example of a generic triangle definition in plane. (b) Example of isosceles right angle triangle definition. (c) Example of internal regular discretisation pattern for an isosceles right angle triangle. (d) Example of unoptimal discretisation pattern.

Additionally, for a unique localisation in the planar space, the coordinates pair (x, y) of at least one vertex need to be known, together with a rotation angle α with respect to the Global Reference System (GRS). By convention, this rotation angle is defined as the angle formed between the y-axis of the GRS and the (directional) vector $\overrightarrow{PA}$ joining vertexes P and A, as shown in Figure 5.4-a.

However, because of the geometrical regularities that one might visually observe in the bispectra of both loading and structural response, simplified triangle configurations, i.e. isosceles triangles (see Figure 5.4-b) are chosen in place of the more generic one. Indeed, this is a choice driven by the need and will of simplifying the process of domain discretisation, which is yet permitted by the problem specificities.

5.2.1.2 The rectangular zone

Together with triangles, rectangles have been chosen as the other basic shape that meshing zones can take. A rectangle could certainly be made up from the close union of two triangles. Nevertheless, explanations and motivations on why such choice has been discarded will be given in the following.

In fact, most of the 2D frequency domain when looking at both the loading and response bispectra can be directly subdivided into many rectangular-shaped subparts, mainly centered in the most critical areas, where the gradient is important, i.e. the quasi-static peak for the bispectra of loading, and the resonant peaks for the bispectra of structural responses.

Considered the likelihood of having an important amount of rectangular shaped zones in the process of their ensemble definition, allowing having their own dedicated representation would avoid mainly two issues, that if repeatedly committed, could lead to a significant loss of performance. The first issue appears evident by

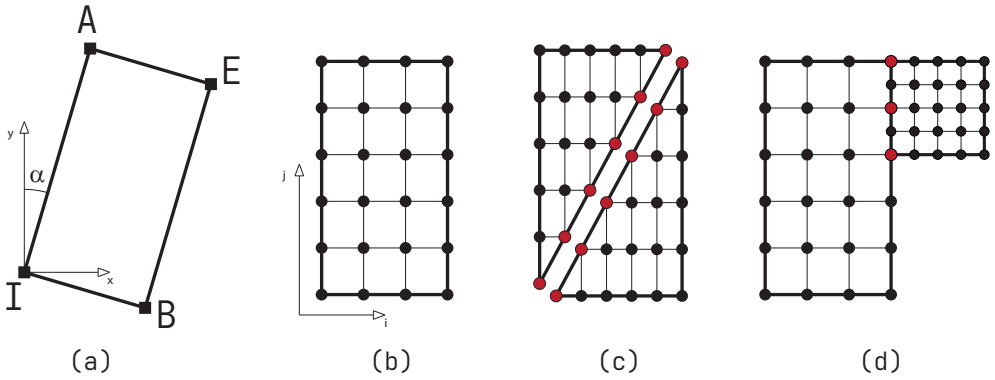


Figure 5.5: (a) Example of a generic rectangle definition in plane. (b) Example of rectangle internal regular discretisation pattern. (c) Example of rectangle internal regular discretisation pattern, if using 2 triangular shaped zones. (d) Example of two adjacent rectangular shaped zones, sharing some discretisation points.

looking at Figure 5.5-c: because one of the main characteristics of a zone is that it is independent from all the others once it is uniquely defined and located, the use of two triangular zones to represent the equivalent rectangle would cause a repetition of discretisation points lying on the two coincident hypotenuses, coloured in red in Figure 5.5-c. Nonetheless, if such repetition could not be avoided for adjacent zones sharing some discretisation points (see Figure 5.5-d), avoiding such points duplication for the internal discretisation of rectangular zones would certainly help having a much more effective algorithmic structure. The other reason lies on how a triangular zone (e.g. isosceles right angle triangle) needs to be internally discretised in order for the algorithmic points of strength could still hold. In fact, because of the cut of the hypotenuse (with respect to the equivalent rectangular zone) such shape would require an equal amount of discretisation points lying on the two smaller sides catheti, see Figures 5.4-c and 5.5-c, hence this symmetry results in the same number of discretisation points falling exactly on the hypotenuse, thus generating an actual regular meshing. Forcing a different amount of discretisation points in the case of a triangle would cause the generation of an uneven internal discretisation pattern, which would invalidate the condition of a simple reconstruction policy without the need to actually save the discretisation points themselves (see Figure 5.4-d).

Under these conditions, the number of discretisation points of a triangular zone is

$$N_{p_{\text{triang}}} = \frac{n^2 + n}{2} \tag{5.1}$$

while for the rectangular one

$$N_{p_{\text{rect}}} = n_i * n_j \tag{5.2}$$

where n_i (n_j) is the number of discretisation points along the side parallel to the local i-th (j-th) axis, $n = \max(n_i, n_j)$ for the triangular zone.

For example, assuming that $n_j > n_i$, the number of points in a triangle-shaped zone is $N_{p_{\text{triang}}} = \frac{{n_j}^2+n_j}{2}$, and the number of discretisation points of a rectangular zone resulting from the combination of two triangular zones is

$$\overline{N_{p_{\text{rect}}}} = n(n+1) = n_j(n_j+1). \tag{5.3}$$

The resulting discretisation points oversampling rate is $\frac{\overline{N_{p_{\text{rect}}}}}{N_{p_{\text{rect}}}} = \frac{(n_j+1)}{n_i}$. Indeed, at best it will hold a unitary value, meaning that either using a single rectangular zone or two triangles would result in the same amount of discretisation points, which is possible if and only if $n_i = n_j$ in the rectangle discretisation. At worse, this oversampling rate can reach values of orders of 10 to 50. Assuming an average oversampling rate of 2 to 5, multiplying this rate by the total number of rectangular zones $\mathtt{N_{rz}}$, which can easily reach values of hundreds or thousands for the most complex cases, the total number of oversampled discretisation points can quickly reach orders of hundreds of thousands. This is clearly an unwanted side effect, considered that for each of these oversampled discretisation points, a highly demanding mathematical operation concerning the projection of a big 3D tensor of nodal loads bispectra onto the structural modal basis has to be performed. It is important to keep in mind that this modal projection constitutes the most expensive numerical operation in the context of (bi-)spectral analyses.

Moreover, the number of additional duplicate discretisation points added to those already used when considering the possible sharing between adjacent zones, as shown in Figure 5.5-d, is equal to

$$n_{\text{dup}} = \sum_{k=1}^{\mathtt{N_{rz}}} \max(n_{i_{\text{k}}}, n_{j_{\text{k}}}). \tag{5.4}$$

As the problem size increases, and the required number of rectangular zones $\mathtt{N_{rz}}$ increases accordingly, the additional number of duplicates can reach significant amounts, of even thousands of points, which, considering the workload associated with the 3D projection of one bispectrum, could considerably degrade the overall algorithmic performances.

Overall, the coupling effects of Equations (5.3) and (5.4) should explain and justify why rectangular shaped zones have been introduced as a base shape type instead of splitting them into two equivalent triangles.

5.2.2 Zone localisation: "influence lines" and "influence areas"

Previous sections discussed the concept of zone, and how it is defined in the context of the MESHER algorithm. Moreover, the basic shapes that each zone is allowed to take in order to ensure the simplest possible algorithmic complexity have been detailed.

Nonetheless, together with how to define a zone, it is necessary to also establish an optimal criteria aimed at defining how to localise each zone within the global domain, i.e. the 2D frequency space in the case of bispectra.

Figure 5.3 shows all 6 possible peaks configuration, for a pre-determined set (triplet) of modal frequencies $(\omega_i, \omega_j, \omega_k)$. Nonetheless, these 6 combinations are to be drawn for all possibles $\omega_i \quad \forall i = 1, \ldots, \mathtt{NM}$ natural frequencies of the NM retained structural vibration modes (see Section 3.1). Normally, for large-scale projects, when performing a modal base analysis, one would usually keep some dozens to few hundreds of vibration modes, so that in the most general fashion there would be $C_3^{\mathtt{NM}}$ of peaks configurations. In these circumstances, the number of all possible configurations that could be drawn as in Figure 5.3 grows very fast with NM.

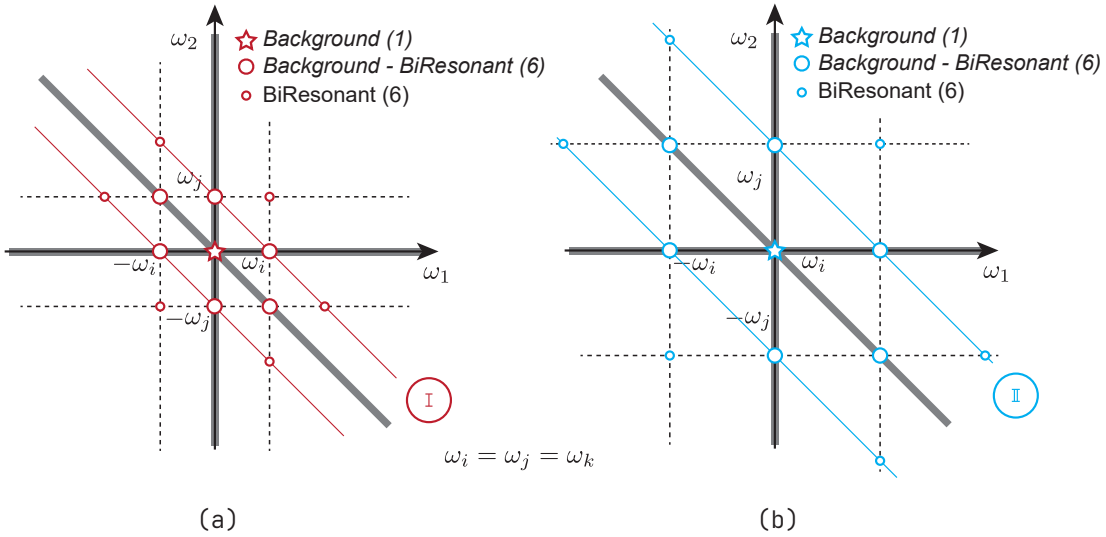


Figure 5.6: Example of same peaks configuration, different vibration modes (i.e. modal frequencies).

Figure 5.6 shows a graphical example of one of the 6 combinations of Figure 5.3, for two different values of modal frequencies. Each dashed line in Figure 5.6 represents an influence line, drawn based on the spatial location of their relative primary resonant peak. Repeating this operation for all resonant peak present in a bispectrum would define the whole set of influence lines, referring to that

specific bispectrum only. Finally, determining the set of influence lines for all bispectra finally leads to the determination of the whole set of influence lines that characterise the specific problem, i.e. structural system in this specific case.

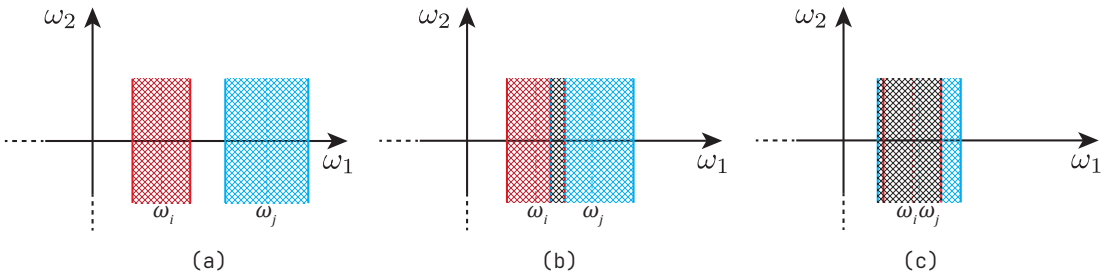


Figure 5.7: Determination of influence areas from influence lines. (a) No overlapping; (b) Partial overlapping; (c) Complete overlapping.

Nonetheless, influence lines alone are not sufficient for a proper and efficient solution of the problem, that is, determining an optimal discretisation of the two-dimensional frequency space. However, if to each influence line an *influence area* is associated, then a finite set of influence areas can indeed fully represent a 2D space. There might be several ways in which the extension of each influence area can be determined. For instance, the simplest would be to define a fixed extension to be applied to all influence lines' influence areas. However, this might not be the best solution, specially in terms of efficiency and accuracy. In order to determine the influence area extension so to balance both accuracy and efficiency, the extension L_i is computed based on the relative mode of vibration's resonance peak width w_i:

$$L_i = \text{cst } w_i \tag{5.5}$$

$$w_i = 2\xi_i\omega_i \tag{5.6}$$

where cst is a tuning constant (freely definable by the user), ξ_i and ω_i are the damping and natural frequency of the structural system in its i-th mode of vibration.

It is clear that, from Equations (5.5) and (5.6), an influence area extension will be larger for modes at higher frequencies, but also for those modes for which a specific (and higher) damping ratio is targeted.

This difference in influence area extensions can lead to three possible scenarios, depicted in Figure 5.7:

(a) the influence lines are far apart and there is no overlapping between the two influence areas;

(b) a partial overlap exists between the two influence areas. The left-border of the influence line at a higher frequency value goes below the right-border of the influence line at the lower frequency;

(c) full overlap between the two influence areas. The left-border of the higher-frequency influence line goes below the left-border of the lower-frequency influence line.

In both cases (b) and (c), the resulting influence area will be the overlap of the base influence areas, the black coloured filling in Figures 5.7-b and 5.7-c.

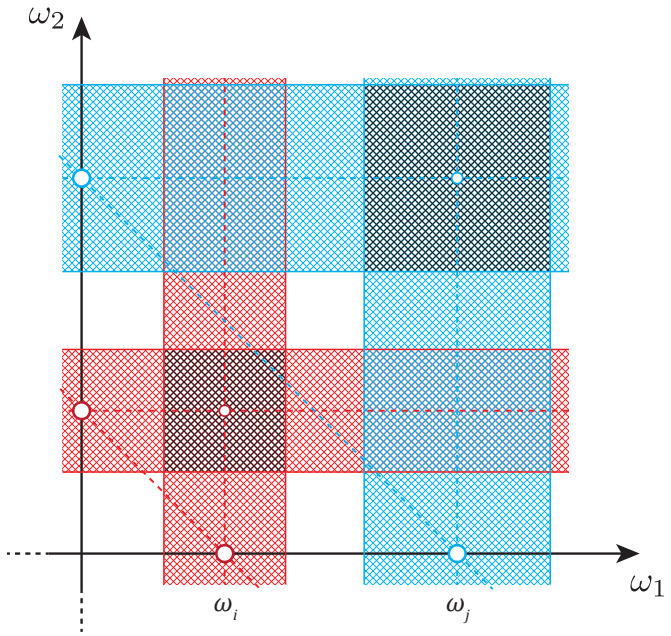


Figure 5.8: Example of how intersection of influence lines defines meshing zones, for case (a) of Figure 5.7. The resulting peak zones are those with shaded filling.

By repeating this process for all set of influence lines and for both spatial directions x and y, the resulting intersections between influence areas will automatically define the MESHER zones. Figure 5.8 gives an example for two influence lines, in the case (a) of Figure 5.7.

5.2.3 The MESHER as an optimal discretisation scheme

5.2.3.1 The "regular" scheme

The simplest and most direct meshing approach would be assuming a regular meshing across the whole domain. That is, defining a finite delta $\Delta\omega$ small enough with which creating a regular 2D-grid of points covering the entire problem domain, as Figure 5.9-a shows. It would then provide the points (ω_1, ω_2) where to compute the loading and response bispectra, as of Equations (C.1), (3.45) and (3.46). This approach clearly breaks the point of avoiding unnecessary as well as

expensive discretisation, specially at the level of bispectra of the loading. Indeed, the chosen grid step $\Delta\omega$ needs to be small enough in order to properly capture those zones where the gradient of the bispectra changes rapidly.

The choice of the grid step $\Delta\omega$ is obviously governed by the most unfavourable case. Nonetheless, this approach might be adopted for very small case studies, for which the problem sizes would result in a fairly economical computation expense (in terms of running time) even ensuring a sufficiently small grid step $\Delta\omega$. However, as soon as the problem size increases, this direct approach would start to suffer significant computational time increases.

5.2.3.2 The "adaptive" scheme

Another possible solution is using an adaptive meshing scheme. By adaptive is meant that discretisation points are added, either randomly or following a localisation scheme, until a given criterion is met. Figure 5.9-b shows a graphical example. This approach is potentially much more efficient than the regular mesh previously introduced. In fact, if one chooses both a proper localisation method and the right criterion (e.g. policy) which drives the convergence of the refinement process, this approach can result in a lower amount of discretisation points while providing at the same time a more representative and accurate discretisation pattern. However, after careful reasoning, one can easily realise that this approach, though promising, and surely powerful in many fields of applications, is not very suited for bispectral analyses. In fact, this discretisation scheme would have to face three major issues:

storage: The adaptive scheme requires to keep track of all the 2D frequency pairs (ω_1, ω_2) for which a discretization point is added during the refinement.

efficiency: The convergence criterion must be checked at every iteration, for each added point. This inevitably adds lots of operations to be executed, which might lead to considerable algorithmic overhead, hurting overall performances. Moreover, since each point might be added in a random fashion, the determination of the set of encircling points against which asses the convergence criterion might itself require a considerable amount of computational effort.

complexity: Another consequence of the random nature of each point localisation results in hard to determine influence areas (where in this case the concept of influence area is not the same as that discussed in Section 5.2.2) to attribute to each point (ω_1, ω_2), when integrating the bispectra to compute the statistical moments. In fact, if this remains relatively easy at second order for a 1D frequency space (i.e spectral analysis) where all points lie on the same line, this process becomes cumbersome as soon as the domain

extends to higher order spaces, e.g. planes, volumes etc. This aspect also influences overall efficiency.

It remains important to note that, as for the regular meshing scheme, for small problem sizes one could easily use the adaptive scheme without noticing any hurt in performances, since all the mentioned drawbacks would be limited. Nonetheless, these aspects would start to manifest their impact as soon as one considers relatively big case studies. To illustrate this aspect quantitatively, the number of discretisation points to be saved could easily exceed the order of 10^6 ($\times 2$ because they would carry a pair of frequencies (ω_1, ω_2)), and for which the convergence criterion should be verified.

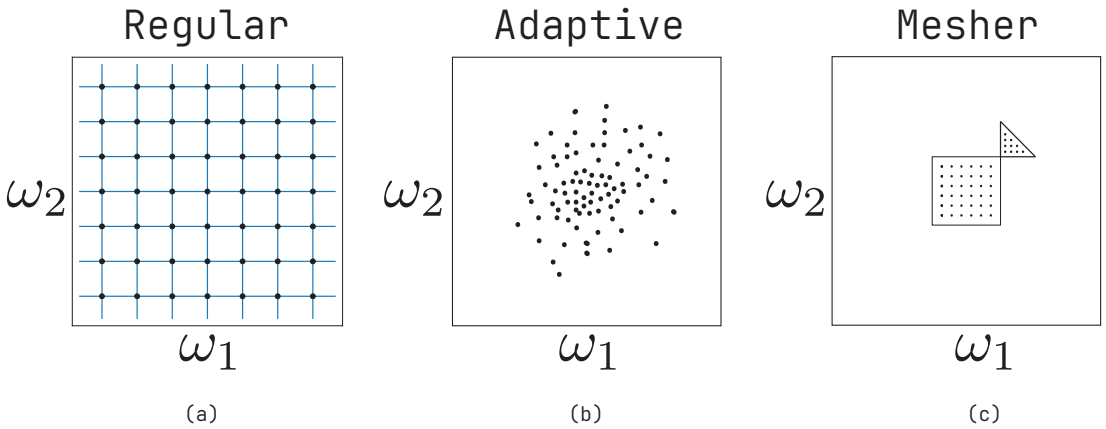


Figure 5.9: (a) Regular 2D mesh grid. (b) Adaptive scheme. (c) Meshing zones.

5.2.3.3 The MESHER scheme

The MESHER is essentially based on a very simple concept: patching the 2D frequency space by means of independent, elementary zones. The concept of a MESHER zone has been discussed in Section 5.2.1.

In simple terms, the MESHER scheme can be seen as a combination of both the regular and adaptive schemes. The adaptive is reflected in the accurate and thoughtful positioning of such zones so that they correctly and optimally cover the areas of a given common morphology (i.e. peaks, crests, basins, flat, etc.). Then, the regular meshing scheme is used within each zone, but in a way that it maximises the actual location of the zone it is going to discretise.

This also allows a clear separation between zone localisation, hence global domain sub-division, to its actual discretisation, which is then done via each zone's internal discretisation policy, completely independent from all other zones. Hence, in a first step, the global domain is split into the set of zones. Then,

once the set is completely defined, the actual definition of the zones' internal discretisation can be deferred to a later stage, without loosing any important information. This configuration, where zones localisation and zones' internal discretisation definitions are two separate concepts is extremely useful in cases where, for instance, the definition of each zone internal discretisation depends on some conditions that need to be verified during computation, but which do not affect their localisation in the global domain. In this framework, the first step consisting of determining the location of each zone can be done regardless of the validation of such conditions. This algorithmic configuration is extremely effective in the light of new researches carried on the hardware acceleration of distributed memory communication frameworks with dedicated network hardware facilities, where some independent computations can be interleaved with network communications, increasing the computational efficiency [90, 91].

5.2.4 The two-stage discretisation process

The introduction of the MESHER and its zones allows for more flexibility in the discretisation process. As discussed in the previous section, each zone can be placed anywhere in the 2D frequency domain, as long as it does not interfere with any other zone, up to the point in which the ensemble of zones patches cover the entire 2D frequency domain.

Also, the conceptual separation between zone localisation and zone discretisation has facilitated the implementation of a two-stage discretisation of the 2D frequency space.

5.2.4.1 First stage: Pre-mesh

Once the ensemble of zones has been identified, the **first stage** consists in a first optimal internal zone discretisation aiming at optimally representing the bispectra of modal loads (see Equation (3.46)). This task is achieved by imposing much stricter internal discretisation policies to those zones in the neighborhood of the origin $(\omega_1, \omega_2) = (0, 0)$ where the bispectra of buffeting loading has large gradients. For each of the points defined in this first discretisation step, bispectra of modal loads are computed. Once done, all the information is saved, together with the reduced data needed to be able to fully reconstruct each zone in a second phase of the computation. In fact, this represents the minimum amount of data that needs to be saved in order for it to be usable in the following steps of discretisation of structural responses bispectra. This first phase is also called the pre-meshing phase, where all the zones covering the entire 2D frequency domain are determined and a preliminary computation of the bispectra of modal loads is done.

5.2.4.2 Second stage: Post-mesh

Once the pre-meshing phase is completed and data is saved, the **second stage** of post-meshing can then take place. It is named *post-mesh* since this meshing stage builds upon the already existing one determined in the first pre-meshing stage. The reason why data is saved, and not actually used on-the-fly (even though this is actually possible) is to have the possibility to resume this second step of the computation at any moment in time, even after having actually stopped the computation right after the pre-meshing phase. As discussed, the most expensive operation when it comes to bispectral analysis is the projection of the 3D matrix of bispectra of nodal loads (see Equation (C.1)) which has dimensions of `NDOFs` × `NDOFs` × `NDOFs`, onto the modal basis to recover the 3D matrix of modal loads bispectra (see Equation (3.45)). Avoiding to repeat this expensive computation would save lots of computational time, specially in tasks such as post-processing for data visualisation, as well as reducing disk memory required to store this big amounts of data.

After projecting, in the post-meshing phase the modal response bispectra need to be determined and integrated, to recover the 3$^{\text{rd}}$ statistical moments, using Equation (3.48). For the reasons explained at the beginning of Section 5.1, many integration points are required in the neighborhood of the resonance peaks, in order to ensure an accurate estimate of the statistical moments of structural responses. On the other hand, due to the different topology between loading and responses bispectra, all the zones that are placed in the neighborhoods of the resonance peaks will have a fairly loose internal discretisation resulting from the first pre-meshing stage, since this stage only focuses on accurately represent the loading bispectra. In that, the MESHER algorithmic configuration allows for a relatively simple and efficient *refinement* process. The following section explains how this is achieved.

5.2.5 The Tail-Head-Previous-Current (THPC) method

In Section 5.2.1, the characterisation of both triangular[2] and rectangular shaped zones has been detailed. Their introduction, within the MESHER discretisation scheme (see Section 5.2.3.3), has been justified, compared to the other considered approaches (see Sections 5.2.3.1 and 5.2.3.2) in the need of finding the best solution in terms of memory occupancy and computational time (CPU time).

Indeed, from a memory management point of view, the conventional regular meshing approach (discussed in Section 5.2.3.1) is the best, since very few information is actually needed in order to fully characterise the entire discretisation pattern. In fact, once the 2D domain boundaries and the grid step in the two

[2]In such context, triangular makes implicit reference to isosceles right angle triangle.

main directions $(\Delta\omega_1, \Delta\omega_2)$ are defined, the full discretisation pattern is uniquely identified. Additionally, when adopting such an approach applied to bispectral analyses, the mesh is ortho-tropic which means that the domain limits and the grid step are equal in the two main domain directions. In such cases, the information even reduces to two domain limits (lower and higher bounds), plus the desired grid step $\Delta\omega$. From a numerical point of view, this translates into 3 floating point numbers, for a total of $3*8 = 24$ bytes, assuming that each one of them requires 8-bytes of memory, equal to 64-bit in common 8-bit processors, for their double precision representation [92]. Moreover, if in place of the grid step $\Delta\omega$ one stores the desired number of discretisation points (per side) $n > 0 \in \mathbb{N}$, the total memory footprint would reduce to $2 * 8 + 4 = 20$ bytes. Then, quite intuitively, the resulting grid step is given by the total domain extension divided by the total number of segments, that is $\Delta\omega = \frac{|\omega_{\max} - \omega_{\min}|}{n-1}$. Nonetheless, if it is extremely optimal in terms of pattern representation memory footprint, it is quite the opposite in resulting computational expenses. In fact, because of the need of ensuring an adequate accuracy, this approach would enforce a very strict policy (i.e. a very low grid step $\Delta\omega$, or equivalently a very high number of discretisation points per side n), dictated by the high gradient peaks, across the whole domain, resulting in an enormous amount of discretisation points, at which the costly operation of projection of 3D tensors of bispectra needs to be performed.

On the contrary, the adaptive scheme introduced in Section 5.2.3.2 is memory consuming since it stores each sampled point, in order to be able the retrieve the discretisation pattern whose complexity is cause by the gradient guided randomness of the refinement strategy. It is straightforward to see that in this case, a total number of $b = \mathrm{N} * 2 * 8 = \mathrm{N} * 2^4$ bytes is necessary. Since for large scale problems, N can reach values of the order of millions (i.e. $10^6 \approx 2^{20}$), b quickly increases to orders of 2^{24} bytes = 16 MB. While this amount of memory is indeed not a real problem for current memory capacity (orders of some Gb), it is indeed not an optimal scenario, also in terms of algorithmic efficiency. In fact, a related problem for such an adaptive scheme is that, the final number of discretisation points N is unknown, so that a fixed-size pre-allocation of memory is not possible. If this pre-allocation cannot be done, then resizable-arrays must be used in order to accommodate the increasing number of added points. Two main *container* types are used for implementing resizable arrays:

- Lists;
- Array/Vector types.

Indeed, none of them is as effective as the case in which N is known beforehand, allowing for the allocation of the right amount of memory needed to hold the information. In fact, while the first method does ensure no waste of memory at all, traversing a list (i.e. when needing to actually inquire the resulting discretisation

pattern) accounts for a non-negligible overhead. In fact, main memory fetching operations are expensive, of orders of $\approx$ 100 CPU cycles [93]. If this operation has to be done for each element accessed, the slowdown might be considerable. Discussion of this aspect can also be found in Section 1.1.4 of [94]. And chances of *cache misses* are high, since memory is not laid down contiguously. On the other hand, the second suffers two main issues:

(i) high likelihood of memory waste, depending on the growing policy. By memory waste is meant memory that is allocated without ever being used, so limiting the amount of residual memory that can be allocated for actual use. In this regard, the concept of *virtual memory* [95] helps avoiding such memory wastes.

(ii) considerable overhead at every memory re-allocation. In fact, every memory re-allocation requires: (i) allocation of the new bigger memory segment, (ii) old data transfer (copy) to new memory segment, (iii) and finally old memory segment release.

In such a scenario, the MESHER scheme stands in the middle. As for the regular scheme, it requires a limited amount of information in order to fully reconstruct the internal discretisation pattern. In such case however, the simplifications that could be made on the conventional regular meshing, assuming it is determined as being symmetric in the two main domain spatial directions, do not hold. In fact, having such a constraint in the discretisation pattern determination would sometimes be opposed to the optimal scenario that the MESHER aims at reaching. Consequently, basic information for both main (local) directions has to be saved, which translates into 2 (local) domain extensions $(\Delta\omega_i, \Delta\omega_j)$, 2 refinements (n_i, n_j), together with the minimum additional required information needed for uniquely localising a zone in the 2D space of frequencies, which reduces to a rotation α and the pair coordinates (x, y) of a reference point, for a total of $5*8+2*4 = 48$ bytes. In modern x86-64 microprocessor architectures, it is less than a cache line, usually of 64 bytes, which means that such an amount of information can be loaded at once in cache memory and readily used, for an optimal memory fetch to data usage ratio. However, this information has to be saved for each independent zone. Considering that in the most complex case studies, the total number of patching zones can reach the order of thousands (millions would be quite unrealistic), the total amount of memory required to store the base information for a full discretisation pattern reconstruction in the MESHER case would amount to orders of $10^3 * 48 \approx 2^{10} * 48 \approx 50$ kB $= 0.05$ MB < 1 MB. So it is clear how in this case, the amount of information is still relatively higher compared to the conventional regular case, which required only 20 bytes, but yet remains at the same time considerably smaller than the one needed in the adaptive scheme approach. And since the internal discretisation of each zone is based

on the deterministic regular scheme, even if the total number of required meshing zones is not known a priori, this would still guarantee that no memory is wasted, since in such a case, every zone would treat its own information independently from the others, saving it to memory for a later fetching and reconstruction.

However, this optimal discretisation representation pattern would be practically ineffective if not paired with an equally effective way to actually retrieve it, minimising the impact that this reconstruction operation might theoretically have on memory usage. To avoid this issue, a reconstruction scheme called the Tail-Head-Previous-Current (THPC) method has been formulated. It hinges on the use of 4 main reference pointers in the local domain, two per main direction (i, j), where the local domain (i, j) directions emulate the global domain (x, y) ones.

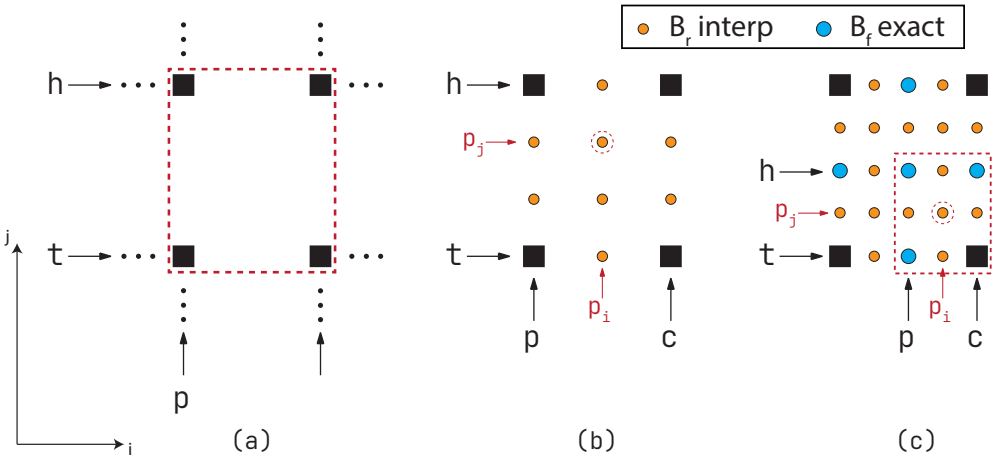


Figure 5.10: Graphical representation of the THPC method. (a) Example of the "sector" determined by the 4 THPC pointers. (b) Example of the two free internal pointers moving within the localised sector. (c) Example of the THPC method applied to a multilevel-refinement method process.

The Head (h) and Tail (t) pointers move along the local j-th direction, while the Previous (p) and Current (c) will instead cover the other i-th direction. They are used to uniquely localise, within the local zone domain, a rectangular shaped sector defined as having its four vertices 4 adjacent discretisation points determined by the initial local meshing established in the first pre-meshing phase. Figure 5.10-a shows a graphical example in the most general case. This phase, as a reminder, is basically the phase in which the MESHER focuses on finding the optimal discretisation pattern to correctly represent the third order spectra at the level of the wind buffeting loading. This is the minimum (and maximum at the same time) required information that needs to be saved by the MESHER in order to properly and optimally be able to operate in further steps, being the

establishment of the optimal discretisation pattern at the level of the structural response. Within the individualised sector, two additional local pointers p_i and p_j are free to move, spanning along and across the discretisation points (coloured in orange in Figure 5.10-b) that needs to be added with respect to the initial discretisation for a better representation of the response bispectra. Clearly, the amount of added points for this stage depends on the internal policy that the zone holds, which depends on the actual area that it covers within the global domain.

All in all, this methodology allows for additional precision, directly at the level of loading bispectra, and so indirectly of structural responses. In fact, in the process of initial discretisation pattern (on-the-fly) reconstruction, if requested by the user, additional discretisation points referring to the correct representation of the spectra at a loading level can be added, even if they were not planned in the first stage. This means that these points are directly computed and not otherwise retrieved. This feature is graphically detailed in Figure 5.10-c, where the additional points are represented by the blue circles, which are then added to the base discretisation pattern used to uniquely identify the sectors in which the local pointers p_i and p_j are allowed to span. This has indeed many advantages. Among all, it allows to clearly separate the desired precision and associated computational times, between the actual estimation of the statistical moments, which needs to be as accurate as possible as discussed at the beginning of Section 5.1, and any possible post-processing data visualisation for which a reduced accuracy is admissible, as long as the obtained graphical rendering do not distort their interpretation from the theoretical behaviour.

5.3 BsaLib: numerical implementation of the bispectral problem

Tackling the bispectral problem of MDOFs structural systems requires a numerical approach. In a first state-of-the-art query, some existing code bases were found [96, 97]. However, all of them tackle the problem from a signal theory point of view, i.e. assuming a time domain, and hence computing the bispectrum as of Equation (2.46). Under a pure stochastic approach, no existing code was found. Therefore, a new development was initiated, that brought to the establishment of BsaLib, acronym of **B**ispectral **s**tochastic **a**nalysis **Lib**rary. The code is publicly available at `https://github.com/miEsMar/BsaLib`. For maximum performances, its core has been written in Modern Fortran[3] [98], a compiled language that together with C and C++ is widely used in the realm of High-Performance Computing (HPC) applications [94].

[3] `https://fortran-lang.org/`

5.3.1 Code & Memory layout

BsaLib has been developed as a plug-in library component, to allow its easy integration in every academic or commercial FEM software. As such, **it requires the hosting program unit to provide BsaLib with the required data in order to function properly.** This external data can be resumed into:

- Data related to the structural system, such as modal matrices, nodal coordinates, vibration modes and frequencies;
- Wind data, such as the type of PSD of wind turbulence, the wind profile type, among all;
- General setting options, to control the behaviour of BsaLib.

For an extended list of data that BsaLib requires, all the BsaLib public Application Programming Interface (API) calls, grouped by macro category, are reported in Appendix D, with detailed explanations.

Also, BsaLib follows the *singleton* design pattern [99]. However, unlike the original formulation, in BsaLib this singleton pattern is practically achieved by having a *stateful* machine implementation, where the correct functioning of the whole infrastructure is provided by a correct setting of an internal and private state. The launch of the main computations of BsaLib should be done in a single-threaded execution, or multi-threaded provided that all other threads do not modify any data to which BsaLib is linked, or issue API calls that might alter the (singleton) state internal to BsaLib.

Figure 5.11 shows a graphical representation of the BsaLib framework infrastructure. The application framework can be split into two macro spaces:

- User space, which is the memory owned by the user hosting program or library;
- BsaLib internal space, which refers to memory directly linked to BsaLib.

Regarding the internals of BsaLib, it can be split into two parts: (i) mutable and (ii) immutable. The mutable part can be set via specialised BsaLib API calls, meaning that the user has no direct access to the internal data. This is not only for security reasons, but for ensuring compatibility and logic control over the data that BsaLib receives from the outside. The immutable part is said so because it cannot be modified via any API call. It however adapts to the mutable part, based on how it is set.

However, the BsaLib memory space is in part tight to the user memory space. This dependency is illustrated by the arrow in the left part of Figure 5.11. The motivation lies in the fact that BsaLib tries to avoid any unnecessary and

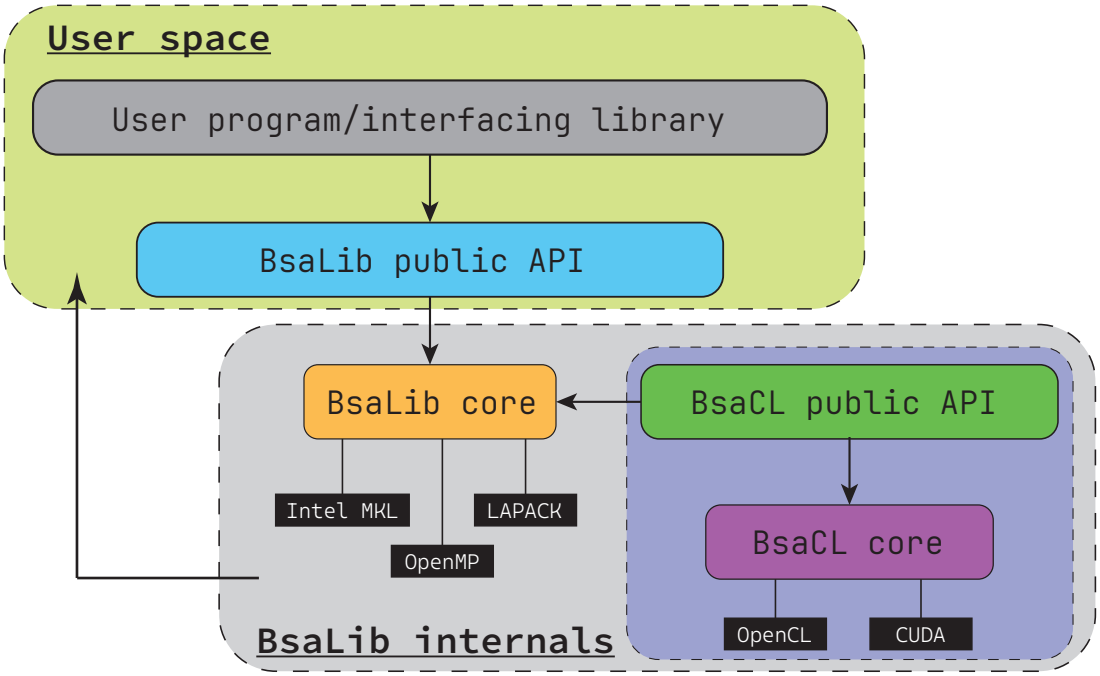


Figure 5.11: Graphical representation of the BsaLib application layers. Top refers to memory space owned by the user. Bottom part refers to memory directly linked to BsaLib.

duplicate memory allocations, not only to reduce the memory footprint of the whole application, but also to increase performances, considered that memory allocations are expensive operations. As a consequence, part of BsaLib memory space directly references user memory space. This is specifically true for non-scalar objects, such as vectors, matrices and tensors.

Looking closely to the BsaLib public API layer, several "categories" of API calls can be defined. They are better illustrated in Figure 5.12. By category is meant to API calls that should be made at a particular stage of the program logic. They are, in order of executing logic:

Initialisation & setup: to ensure a correct and full setup of the internals, BsaLib requires to be initialised (see Appendix D.1.6). After initialisation, several API calls allow the user to set up the various settings, before issuing the main run.

Execution: this stage performs the actual computation (see Appendix D.1.7). If execution is successful, meaning that input data was set up correctly and coherently, results are returned to the user. To minimise the amount of information to be computed and avoid unnecessary costly computations, only statistical information up to third order of modal responses (see Equations

(3.40) and (3.48)) are provided. All in all, this information is the minimum required for reconstructing statistical information of any other structural response information (e.g. displacements, internal forces, stresses, etc.).

Post-processing: BsaLib offers some API calls to perform some basic post-processing of results. In particular, it offers built-in procedures that internally implement, based on the user-specified options, the various kind of modal recombinations techniques, `SRSS` or `CQC` at second order and `CRSC` or `CCC` at third order, as discussed in Section 3.3. This stage is optional, or can be replaced by user-defined procedures which might be better integrated with the hosting environment.

Finalisation: finally, memory needs to be released (see Appendix D.1.8).

When interfacing with BsaLib, this logic ordering in API calls is fairly important. It is crucial to ensure that any API call of a given category cannot happen after an API call belonging to a following stage. At best, such calls would not yield the desired effect (dead calls). At worse, they could lead to the crash of the whole application. In this regard, BsaLib tries to handle all possible combinations of errors related to a wrong order in API calls (among others), leading to an anticipated program termination, with a proper error code and explanation.

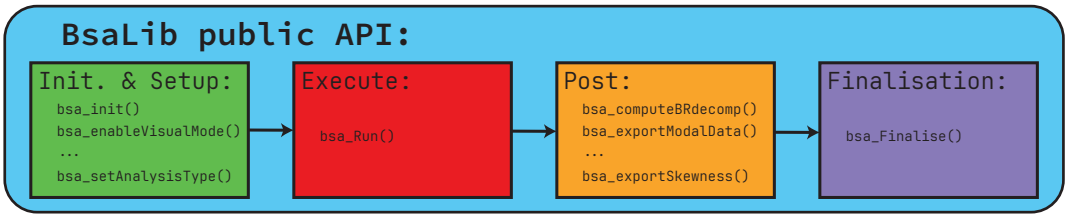


Figure 5.12: Detailing of the BsaLib public API calls. All API calls can be split into several "categories", based on their purpose. Each category's API call must be placed after all API calls of the preceding one, following the logic order that lies between each category.

An example of user program code structure is given hereafter, emphasising the different logical structure when using with BsaLib, as also shown in Figure 5.12:

```
! File: main.f90
program example
    use, non_intrinsic :: BsaLib
    implicit none (type, external)
    ! your declarations here

    ! your logic here

    ! initialise BsaLib
    call bsa_Init()

    ! set BsaLib internal state through its API

    ! once done, run BsaLib
    call bsa_Run( ... args ... )

    ! Eventually, post-process BsaLib results

    ! finally, release BsaLib memory
    call bsa_Finalise()
end program
```

5.3.2 A dual-core implementation

Not only BsaLib strives for optimal performances, but is also meant to be as general as possible. To do so, the internal core of BsaLib is split into two main sub-cores, as shown in Figure 5.13.

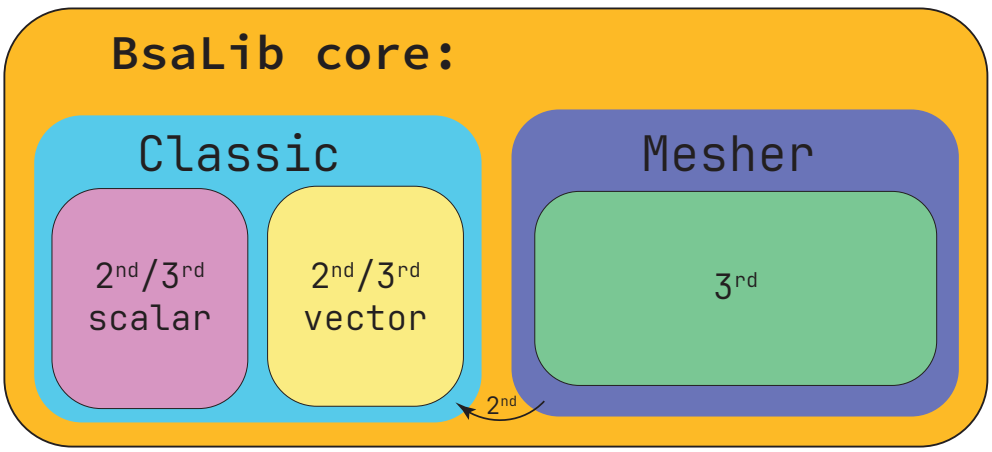


Figure 5.13: Representation of the dual-core implementation. The main hidden core of BsaLib is split into two main sub-cores: (left) implementation of the "classic" approach; (right) implementation of the novel Mesher approach.

Classic: This core gathers the "naïve" implementation of the Spectral and Bispectral problem. That is, equations are implemented assuming a regular meshing of the 2D frequency domain. Details of this discretisation pattern have been given in Section 5.2.3.1. Nonetheless, even if this is a trivial implementation, several numerical optimisation measures have been taken into account into their actual implementation, in particular to ensure memory efficiency. This core is further split into two implementations:

- **Scalar** core implementation, in which equations are implemented for each frequency or pair of frequencies individually.
- **Vector** core implementation, which makes use of explicit vector arithmetic operations (additions, multiplications, etc.) that the compiler can map to a set of vector instructions bound to the underlying hardware [100, 101].

Mesher: This core implements all the algorithmic novelties and optimisations that have been discussed throughout previous Chapters. In this case, unlike the classic approach, there is no separation between a scalar and a vectorised implementation in this case.

While indeed the vectorised approach is preferable for performances, it has memory limitations since it treats the whole set of discretisation frequencies at once. This is specially true for 3D tensor of bispectra, which, from an implementation perspective, end up being a 5D tensor, whose total size can easily reach the order of tens of giga-bytes (Gb $= 10^9$).

The classic core implements both 2^{nd} and 3^{rd} orders, i.e. both Spectral and Bispectral analyses, for both scalar and vectorised implementations. On the other hand, the MESHER implementation, currently supports 3^{rd} order only, hence relying on the classic approach implementation for the 2^{nd} order.

5.3.3 The two-stage `Mesher` implementation

Section 5.2.4 has introduced the concept of the two stages that characterise the MESHER approach. They are:

Pre-mesh: This stage is the most important one. It provides:

1. Definition and localisation of all the meshing zones (see Section 5.2.1);
2. Definition of a first, high-level discretisation of each zone to achieve better discretising modal forces bispectra.
3. Computation of bispectra information at all these discretisation points. This information will be the starting point for the second stage.

4. Saving the computed bispectra. This is made via I/O, writing data to a binary file, since the saving must be consistent and durable, so that data can be retrieved at any point in a future time.

Algorithm 1 shows a high-level overview of the structure of the Pre-Mesh stage.

Post-mesh: This stage complements the Pre-mesh one, extending the discretisation pattern defined in the first stage to better represent modal responses bispectra. This is enabled by the THPC-method (see Section 5.2.5). First, the bispectra information is read back from the binary file. Then, each zone's internal discretisation is refined, and third moments of modal responses are computed on the resulting refined pattern. This is the most important information that needs to be computed, and from which statistical information of any other kind of structural response can be computed. This is why, statistical moments of modal responses is what BsaLib returns to the user.

Algorithm 1 Algorithmic structure of the `PreMesh` function.

```
function PreMesh
    Mesher-specific data initialisation
    nat_freqs ← natural modal frequencies of the structural system
    infl_areas ← computeInflAreas(nat_freqs)
    infl_areas ← resolveOverlaps(infl_areas)
    n_lims ← getNumOfLimitsFromInflAreas(infl_areas)
    bkg_zone ← defineBkgZone()
    computeZone(bkg_zone)
    i_start ← index of first influence area not covered by bkg zone
    if i_start ≤ n_lims then
        for i = i_start : n_lims do
            zone ← defineZoneFromLimitIndex(i)
            computeZone(zone)
```

The distinction of Pre- and Post-mesh stages also allows for a better and clear separation between actual computation of statistical moments, and possibly, a simple post-process visualisation stage. In fact, if in the former case more refined meshing patterns are required for a more accurate estimate of statistical moments, in the latter a looser discretisation policy can be allowed, since it affects visualisation only. This would indeed make post-processing for visualisation much faster than in cases where actual statistical moments are to be computed. A graphical representation of such structure is given in Figure 5.14.

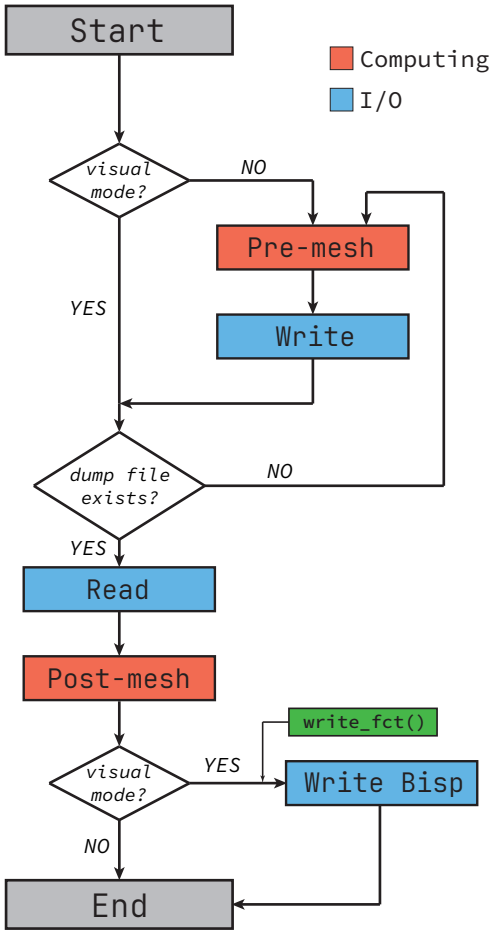


Figure 5.14: Logic flow for the two-stage implementation in the MESHER approach.

If the post-mesh phase is done for visualisation purposes (i.e. `visual_mode` ON), the Pre-mesh phase is skipped, provided that the binary file exists, otherwise it is created by forcing the Pre-mesh phase to be executed. At the exit of the Post-mesh phase, if visualisation is ON, control flow branches to a block where data regarding bispectra of structural responses can be actually used and post-processed. To keep things as general as possible, BSALIB provides a mean to specify a user-specific procedure to be called when entering this branch, so that the user can plug any desired post-processing facility directly into BSALIB (see Section D.7.2.2). This procedure will be then called for each individual meshing zone. If no user-specific procedure is provided, BSALIB still provides basic built-in Post-mesh visualisation functionality, which currently only support writing the requested data to file with the same format used in the Pre-mesh phase, so that it

can be used in an external visualisation framework. Ideally, in future implementations, exporting formats compatible with some of the most used visualisation frameworks [102] could be added.

5.4 Bispectral analysis in the Era of HPC

The formulation of the MESHER discretisation scheme was introduced with the main goal of providing a way to efficiently determine the optimal discretisation patterns of both buffeting loading and structural response bispectra, maximising numerical precision while minimising computational time and complexity. This is achieved by means of a two-stage discretisation process (see Section 5.2.4), where in a first stage each zone internal discretisation is determined for an optimal representation of the loading (bi-)spectra, and in a second stage, it is extended to take into account the effects of the introduction of higher order Volterra kernels (see Equation (3.47)), which affects morphology of (bi-)spectra of resulting structural responses. In addition, this clear separation is also made possible by the meshing zones and their interdependence, with the THPC scheme (see Section 5.2.5), which allows for a flexible internal zone discretisation, easily adaptable to the different requirements that the meshing of loading and response bispectra need.

5.4.1 `Mesher` suitability to scalable parallel computing

Modern era computers inherit from John Von Neumann's idea of electronic computer, back in 1947 [103]. Over the years, technological advance and a higher demand for more accessible supercomputers led to the development of modern parallel systems, both at a hardware and software level [104].

Nowadays, every modern algorithm that is developed from scratch should better account for the existence of all the parallel paradigms to be effective and competitive on a technological level. *Scalability* is a common measure of how well an algorithm adapts to parallel systems [105].

All these considerations have been well taken into account in the development of the MESHER algorithmic arrangement. Specifically, each zone's independence is what makes it suitable for modern parallelisation techniques, which are vastly used when developing highly performant algorithms. The two most important complementary, cross-platform standardised parallel specifications are:

1. *Open Multi-Processing* (OPENMP) [106], which was initially conceived for shared memory architectures [107, 108]. OPENMP follows a *single instruction multiple data* (SIMD) execution model.

2. *Message Passing Interface* (MPI) [109], best suited for distributed memory architectures [110] (i.e. multi-node clusters, servers, etc.). Nonetheless, it might also be used in shared memory architectures. MPI follows a *multiple program multiple data* (MPMD) execution model.

Regardless of the fundamental difference of the two specifications, nothing avoids these two parallel specifications to be used together [111]. Indeed, the final programming model should be well thought based on the problem specificities.

Another substantial difference between these two specifications is that, while `MPI` is fully implemented with a proper API, `OpenMP` follows a *directive* based programming model, where directives act as "suggestions" to the compiler on how to treat the relative code block. These compiler directives are comment lines with a specific format, understandable by compatible compilers. If a compiler that does not support such programming paradigm is used, these directives are ignored, and treated as a normal comment line.

1: *<OpenMP parallel region directive >*
2: **for** $i = 1 \dots N$ **do**
3: $\quad i -$ *independent instructions here ...*
4: *<OpenMP end parallel region directive >*

Listing 5.1 shows an OpenMP example of summing two vectors in parallel, for the Fortran language. The runtime then distributes the execution among a set of *threads*, which are the base elements in parallel executions on shared memory architectures.

Listing 5.1: Fortran example of OpenMP parallel vector sum.

```
!$omp parallel do
do i = 1, N
   c(i) = a(i) + b(i)
enddo
!$omp end parallel do
```

Both OpenMP and MPI parallel paradigms are particularly suited in those cases where one iteration does not depend on the outcome of previous ones. However, this optimality requirement does not apply to all scenarios, so that, in general, some sort of synchronisation between execution cycles must be implemented in order to guarantee correctness of results.

The proposed Mesher algorithmic structure can potentially benefit from both. In fact, once the whole ensemble of meshing zones is uniquely determined, such ensemble can be then split into an equally balanced set of subgroups, where the number of subgroups is dictated by the number of available MPI processes

(i.e. machines) in the distributed system, or cores if employed in a shared memory environment.

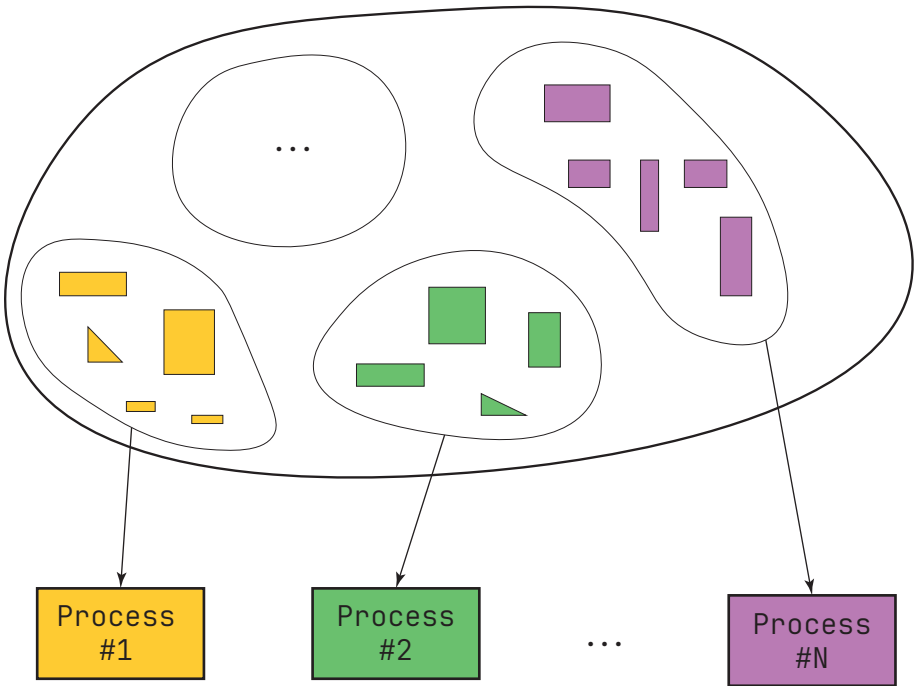


Figure 5.15: Graphical example of splitting MESHER zones into balanced subgroups, in a MPI compatible algorithmic implementation. Each subgroup will be independently scheduled onto a single MPI process.

Figure 5.15 shows a graphical representation of such zones subsets distribution across N processes, where each process could be seen as an independent machine.

Then, within each subgroup, OPENMP parallelism could be used for each zone's internal discretisation process, done using the THPC-scheme, discussed in Section 5.2.5.

This enhanced parallelisation would considerably reduce the total required computational time, specially when applying bispectral analyses to very large (MDOFs) systems, as well as guarantee an optimal usage of memory without too much overloads. Nonetheless, for small to medium sized problems (around a thousand DOFs), even a single machine usage would keep computational time within the order of a few hours at most.

5.4.2 GPU Offloading

Graphics Processing Units (GPUs) are specialised processors that were initially developed to accelerate graphics rendering and geometry transformations [112], using specialised pipelined computations [113, 114].

Whereas Central Processing Units (CPUs) are designed to extract as much parallelism as possible from sequential programs, modern GPUs, also referred to as General Purpose GPUs (GPGPUs) [115, 116], are designed to efficiently execute explicitly parallel programs, specially for programs that inherently have *data-level parallelism* [117]. GPGPUs follow the *single instruction multiple thread* (SIMT) execution model, where the (SIMD) model is combined with *multithreading* [118].

For everyone willing to approach GPU programming, the first question that might arise is: how, and with which tools? Since GPU programming paradigms are fundamentally different with respect to those most suitable to CPU parallel programming, it is crucial to select the right tools and to properly understand how to use them to get the most performance benefit out of them. However, transitioning from the most common CPU parallel paradigms to the new GPU programming and memory models might be a significant challenge.

The two dominant GPU programming models established over the years are *Compute Unified Device Architecture* (CUDA) and *Open Computing Language* (OpenCL) [119]. CUDA is a programming and memory model [120] developed by NVIDIA, firstly introduced in 2006 [112]. OpenCL [121, 122] is an open specification standard, originally developed by Apple with its first release in 2009, and maintained by Khronos Group [123]. Nonetheless, trends from recent years show that SYCL programming model is gradually replacing OpenCL [124, 125]. Also, a programming model inspired on the OpenMP directive-based approach, named *Open Accelerators* (OpenACC) [126] has been developed to overcome these programming model complexity issues. Though such a programming model is very easy to use, specially to those already familiar with OpenMP, it is not widely supported by all major compiler vendors. Additionally, due to its consistently increasing popularity, OpenMP has started to implement offloading directives to directly integrate heterogeneous computing in its programming model [127]. Although these directive-based GPU programming models are quite easy to use while still delivering good performances improvement specially in simple scenarios, they have not been included in the development of the GPU offloading capabilities of BsaLib. On the other hand, BsaLib supports both CUDA and OpenCL programming models in its core implementation. This choice has been made in the light of the fact that, even though both OpenACC and OpenMP are very powerful tools, more lower-level APIs such those provided by CUDA and OpenCL specifications must be used to get the maximum performances from these acceleration devices.

In the context of this Thesis' work, GPU computing has been investigated and implemented to tackle and solve the main computational bottleneck of Bispectral analyses: the multi-dimensional projection of the 3D-tensor of nodal loads bispectra onto the modal base (see Equation (3.45)).

GPU offloading in BsaLib is implemented as an optional plug-in, which must be included at compile-time via a dedicated compile-flag. This is done for maximum flexibility in code compilation, leaving space for automated GPU support detection, and consequent switching-off of the feature whenever a suitable GPU processor is not available in the target architecture. If a GPU processor is available, then GPU offloading can be effectively enabled at runtime, by invoking a dedicated API call (see Section D.1.2). This plug-in is shown coloured in blue-sky in Figure 5.11.

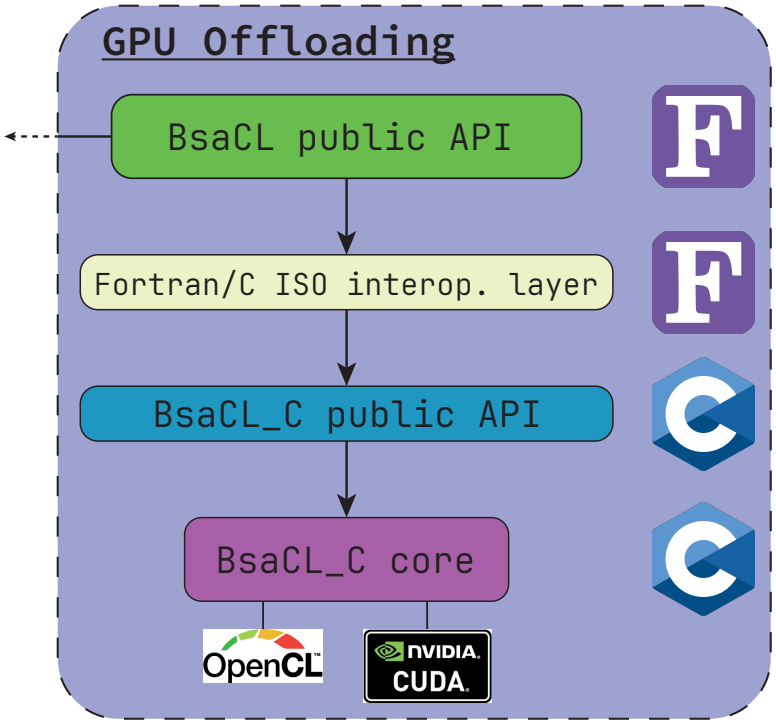


Figure 5.16: GPU Offloading BsaLib code layers.

Figure 5.16 shows a detailed view, where its internal layers are shown. These layers serve as a stable, portable, and standard-conforming transition from the Fortran programming language, in which BsaLib was originally written, and the C programming language [128], from which both the CUDA and OpenCL languages have been developed [113, 122]. Specifically, the BsaLib GPU Offloading integration consists in several layers:

Fortran public API to the BsaLib core: this layer provides direct access to the BsaCL GPU C-core library from Fortran program unit entities, abstracting the transition between the two language processors.

Fortran-C transition layer: this layer implements the transition from a Fortran calling side to the underlying C core library, using standard-conforming processor implementations [129, 130].

`BsaCL_C` **public API:** this layer exposes the public API of the GPU C-core library.

GPU C-core library : this last layer implements the main core logic of GPU integration, and directly integrates the lower level API of both CUDA and OpenCL.

Figure 5.17 shows how the computational domain is split, before issuing the execution of the GPU kernel code. The whole domain is taken as a 2D $\mathtt{N}_\mathtt{f}^2 \times \mathtt{NM}^3$ set of computational elements, split into a 2D grid of work-groups (or thread-blocks in CUDA), each one of them split into a set of $32/64/128 \times 1$ work-items (or threads in CUDA), which are assumed to be the basic execution elements from a hardware point of view. Nonetheless, considering common GPU architectures, GPU's schedulers are built to issue at least 32 threads at a time, what's in CUDA terms is called a *warp* [131, 132, 133], reason for which it is highly suggested work-groups of at least 32 work-items, or multiple of 32 otherwise. It is indeed possible to have work-groups with less than 32 or a number not multiple of 32, which would cause the issuing of an uneven warp, where part of the issued threads will be shut off, reducing performances and throughput of the computation.

Considered the domain decomposition as shown in Figure 5.17, each work-item treats (i) a unique pair of frequencies (ω_1, ω_2) and (ii) a unique triplet of modal indexes (m, n, o) (see Equation (3.45)). Hence, each work-item iterates over all possible combinations of nodal indexes (i, j, k). The choice of selecting as 2^{nd} dimension the modal extent instead of the nodal, relies on how the memory is managed and shared between work-groups. Figure 5.18 shows a graphical representation of the memory model of modern GPUs.

Host memory is represented by CPU memory. Like CPUs, GPUs also have their main global memory, which is accessible from anywhere inside the GPU. When offloading computation to GPUs, data must be transferred to GPU's main memory so that it can be accessed by GPU's processors. Conceptually, a GPU processor is what a core is for a CPU, and is what is referred to a *compute unit* (CU) in Figure 5.18 (OpenCL terms, or SM in CUDA terms). Nonetheless, physically they are built considerably different. Finally, each CU is host of several *processing elements* (PEs) (or *threads* in CUDA), which are conceptually similar to CPU threads.

The reason why the second dimension has been taken as $\mathtt{NM}^3$, and not $\mathtt{NN}^3$ for instance, stands in the memory model of GPUs. In fact, memory synchronisation can happen only within the same work-group, while there is no way to synchronise global memory across different work-groups. This is basically due to the fact that

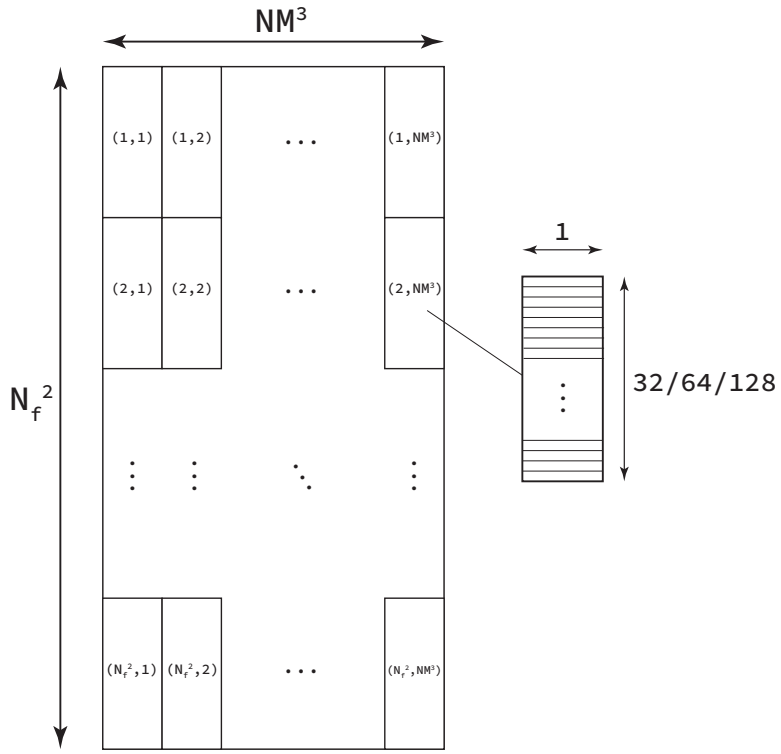


Figure 5.17: GPU 2D-domain decomposition into 2D work-groups, and detailed work-group decomposition into single work-items.

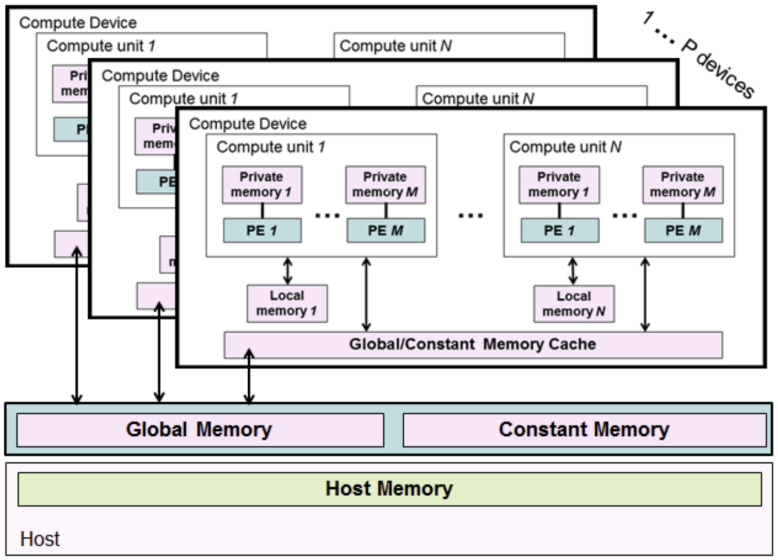


Figure 5.18: Modern GPUs memory model. Image taken from [134].

the scheduler has no requirements on the order of execution of each work-group, and treats them independent from one another‘. Therefore, the only way to share memory between work-groups is in fact to duplicate that memory across each work-group. Only once the computation is done, and data is copied back to the host, it can be processed to get the desired result. For large problems, this issue can lead to a considerable amount of memory waste, certainly unwanted especially when dealing with GPUs memory, which is less available than CPUs ones.

Chapter 6

Applications

In this Chapter, several applications of the bispectral analysis will be presented.

Before proceeding with conducting bispectral analyses on more complex applications, it is important to mention that during the whole development phase, until this final stage, the algorithmic structure has been continuously tested and validated against more conventional approaches. Specifically, the novel bispectral analysis implementation of the proposed framework (MESHER algorithm + POD formulation) has been compared to two different approaches:

- Bispectral analysis following the direct application and implementation of the mathematical equations (see Equations (3.45) to (3.48)) on a regular mesh (see Section 5.2), labelled as "classic";
- Monte Carlo simulations in the time domain, by directly solving the equations of motion in Equation (3.1).

The first comparison was certainly useful to prove that, within the same context (stochastic approaches) but using two different numerical approaches (MESHER and "classic"), results were identical, proving the stability and reliability of the novel algorithmic arrangement compared to a brute and direct application of the mathematical concepts.

The second validation, against Monte Carlo simulations (in the time domain), was also a crucial aspect to consider in order to prove the validity of spectral approaches, in a Gaussian and, more so in a non-Gaussian framework. When considering the non-Gaussian nature of the wind load, if a time domain approach is desired, great care should be adopted regarding the temporal signals used as input. In fact, when higher-order statistics are to be captured in a discrete signal, measurements (or digital simulations) must be long enough in time to correctly capture the statistical information, until convergence is reached. Shorter recordings will inevitably lead to computed statistics which will be highly dependent on

the sample used for the analysis. Spectral and bispectral analyses provide instead a stochastic approach, which is much more reliable once a proper probabilistic description of the input is given, in the sense that a second run with the same set of parameters would provide exactly the same results.

To do so, a test benchmark has been set up, simple enough not to require too much of computational time, but at the same time to be as complete as possible in order to highlight some important aspects, specially in a non-Gaussian context. Figure 6.1 shows a schematic representation. It consists of a 3-DOFs cantilever column, with idealised lumped masses concentrated at the level of the idealised building's floors to model the actual building's mass.

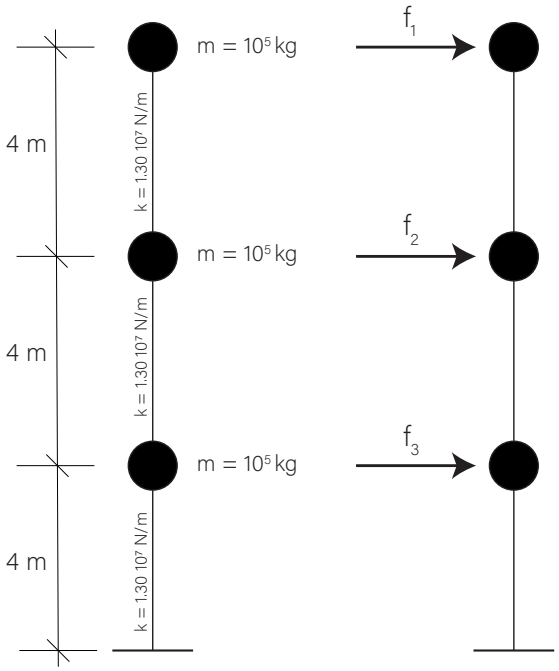


Figure 6.1: 3-DOF system configuration used to validate the MESHER algorithm against conventional approaches.

Extensive information about the generation of wind data, analysis configuration and results are given in the various technical reports that have been delivered throughout the duration of the FINELG2020 Research Project [135] summing up to almost a total of 200 pages [136, 137, 138]. In these reports, the MESHER approach is validated against the "classic" and Monte Carlo Simulations approaches, with the three approaches showing identical results.

As illustrated in the introduction, these two methods ("classic" and Monte Carlo simulations) require a significant number of sampling points for an accurate estimation of higher moments. They could not be implemented in the

following examples having many more degrees-of-freedom than in this simple 3-DOF structure. The discussion is therefore mostly based on how the results are obtained, and on the possible recombinations.

6.1 Quetzalapa Bridge: a simplified model

This section covers an academic application, though based on an existing structure. In fact, in this case, the interest was in applying the proposed methodology to a flexible structure, with a simplified structural model, but for which some wind tunnel testing was done, and aerodynamic wind coefficients (see Equation (4.9)) were available. The choice has been made among bridge decks studied in [139], where the authors compare and classify according to the bridge topologies the wind force coefficients measured for 18 different cross-sections. The choice has fallen on the twin-girder Quetzalapa Bridge, built in Mexico in 1993 on the famous Mexico City-Acapulco highway, after the other famous Mezcala Bridge. It is a two H-tower cable-stayed bridge, with main span of 213 meters, with maximum height of 110 meters.

Figure 6.2: Photo of the Quetzalapa Bridge, Mexico. Source: `http://www.highestbridges.com/wiki/index.php?title=Quetzalapa_Bridge`

Top part of Figure 6.3-a shows the simplified structural scheme. A 3-span simply supported bridge model has been assumed, with main span of 150m, lateral spans of 120m each. Each span has been modeled with 20 beam elements of equal length. The value of the bending stiffness is chosen as $EI = 2.785 \cdot 10^{13}$ Nm^2 and the mass per unit length is equal to 130000 kg/m. The deck width is 29.50 m. On the bottom part of Figure 6.3, the first 4 vibration modes, in the vertical direction, are shown. In total, the model comprises 366 DOFs and seven modes are used in the bispectral analysis. A damping ratio of 3% has been imposed to all vibration modes. In addition to that, the aerodynamic damping has also been

computed with a quasi-steady analysis, and taken into account. Table 6.1 lists the damping ratios for the first 4 vibration modes as well as natural frequencies and modal masses.

Table 6.1: Natural frequencies, damping ratios and modal masses for the first four vibration modes.

mode [-]	Natural frequency [Hz]	Structural [%]	Aero [%]	Modal mass [to.]
1	0.952	3 %	0.45 %	26031
2	1.452	3 %	0.85 %	28973
3	1.788	3 %	0.48 %	38343
4	3.640	3 %	0.07 %	22075

Only the left-most span has been assumed loaded by the turbulent wind flow. Indeed, in deep valleys, the wind flow attacking a bridge deck might be uneven and sheltering effects might result in very unequal loads on the different parts of the deck. This choice has also been made in this context for maximising the asymmetry in the loading resulting therefore in a critical design situation. Besides, with a wind load distributed along a shorter area, the modal loads and therefore the modal responses too, tend to be more non-Gaussian. Indeed, in a final structural design perspective, other load distributions should be considered too.

Figure 6.3-b shows the variation of drag and lift wind coefficients measured in the wind tunnel tests performed on the bridge model, tested at the CSTB, Nantes [139], as well as C_Z, the resultant along the vertical axis in the local element reference system of the drag and lift coefficients, as depicted schematically in Figure 6.3-b. This example has been chosen for the nonlinear nature of the vertical coefficient with respect to the wind angle of attack. In order to simplify the aerodynamic modeling, the vertical force coefficient has been approximated by a quadratic which results in a non-Gaussian wind loading.

More specifically, the illustration focuses on the vertical vibrations of the bridge resulting from this vertical wind load (per unit length), which is defined as:

$$F_Z(t) = \frac{1}{2}\rho C_Z\left[\beta\left(t\right)\right]V^2\left(t\right)B \tag{6.1}$$

where C_Z is the vertical wind force coefficient, measured in the wind tunnel. In a quasi-steady approach, Equation (6.1) translates the influence of the wind turbulence into forces applied to the structure, with which engineers can perform usual static/dynamic analyses. The nonlinearity is hidden behind two factors in Equation (6.1) (see Section 4.4): (i) V^2 which is the square of the norm of the (instantaneous) wind speed vector

$$V^2 = \overline{U}^2 + 2\overline{U}u + u^2 + v^2 + w^2 \tag{6.2}$$

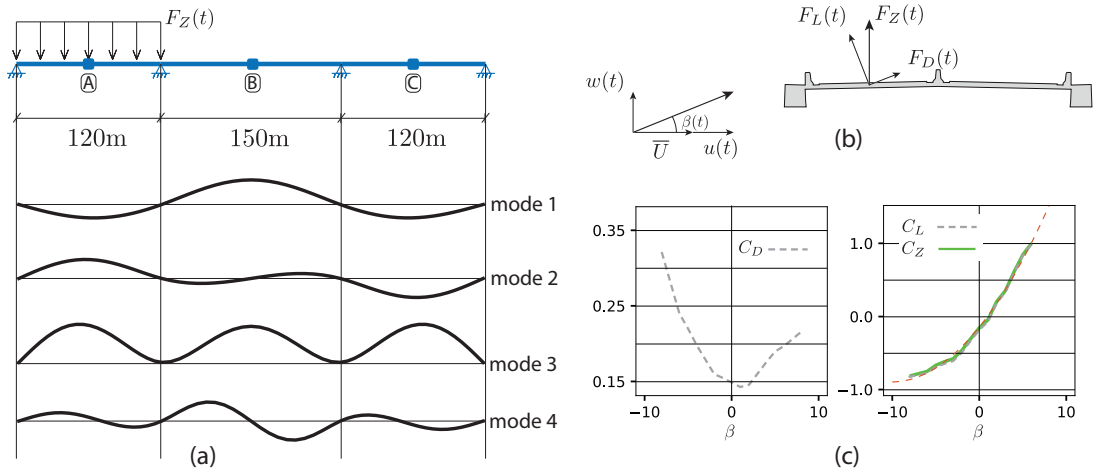


Figure 6.3: (a) Sketch of the studied structure and mode shapes, (b) conventions for aerodynamic loads, (c) considered aerodynamic coefficients.

and (ii) $C_Z = C_Z\left[\beta(t)\right]$ the vertical wind force coefficient, which is a function of $\beta(t)$ the instantaneous incidence of the wind speed, see Figure 6.3-b. In spite of existing more general 3-D models of wind loads [62], lacking modeling information, here a 2-D model is assumed where the wind incidence fluctuates around its mean value $\beta(t) = \overline{\beta} + \Delta\beta(t)$, where $\Delta\beta(t)$ is the instantaneous fluctuation from mean incidence angle at time t. Assuming this deviation to be small, the wind force coefficient C_Z is expressed via a Taylor series expansion around the mean incidence angle $\overline{\beta} = -1°$:

$$C_Z(\beta(t)) = C_Z(\overline{\beta}) + \partial_\beta C_Z(\overline{\beta})\Delta\beta(t) + \frac{1}{2}\partial_\beta^2 C_Z(\overline{\beta})\Delta\beta^2(t) + \dots \tag{6.3}$$

so that $C_Z\left(\overline{\beta}\right) = -0.35$, $\partial_\beta C_Z(\overline{\beta}) = 8.5$ and $\partial_\beta^2 C_Z(\overline{\beta}) = 72.5$, see quadratic approximation in Figure 6.3-c, and

$$\Delta\beta(t) = \arctan\left(\frac{w(t) - \dot{h}(t)}{\overline{U} + u(t)}\right) \tag{6.4}$$

in a simplified scenario where the along-wind structural motion has not been taken into account. Since vertical motion only is considered in this example, the relative angle of attack is computed by subtracting the heaving velocity $\dot{h}(t)$ to the vertical turbulence component.

Substitution of (6.2-6.4) into Equation (6.1), and series expansion for small u, w and $\dot{h}$ yields

$$\begin{aligned} F_Z(t) =& \overline{F_Z} + \frac{1}{2}\rho B\overline{U}\left(2C_Z\left(\overline{\beta}\right)u + \partial_\beta C_Z(\overline{\beta})\left(w - \dot{h}\right)\right) \\ &+ \frac{1}{2}\rho B\overline{U}\left(C_Z\left(\overline{\beta}\right)\left(u^2 + w^2\right) + \partial_\beta C_Z(\overline{\beta})u\left(w - \dot{h}\right) + \frac{1}{2}\partial_\beta^2 C_Z(\overline{\beta})\left(w - \dot{h}\right)^2\right) \end{aligned} \tag{6.5}$$

where $\overline{F_Z} = \frac{1}{2}\rho C_Z(\overline{\beta})\overline{U}^2 B$ is the average aerodynamic load (treated in a separate static analysis, see Section 3.3.1), and the linear and quadratic loading terms are readily identified. In this expression, the argument (t) has been removed to simplify the notation and the transverse component of the wind $v(t)$, parallel to the bridge axis, has also been neglected. By keeping or discarding the second line in Equation (6.5), the quadratic loading terms are kept or discarded. Table 6.2 lists all the relevant data regarding the wind turbulence, having a mean wind speed of $\overline{U} = 38$ [m/s].

Table 6.2: Summary of important wind turbulence properties.

turbulence	Lengthscale [m]	std [m/s]	Coherence coeff. [-]
$u(t)$	250	6.5	12
$w(t)$	200	5.24	12

Concerning the spectral and bispectral analysis, several variants are considered. First, in order to evaluate the importance of the spatial coherence of wind turbulence (and consequently buffeting loads), a variant of the original problem with coherence coefficients for u and w tending toward infinity is simulated. This variant is labelled "D-" in the following Figures, to highlight the fact that the CPSD matrices of turbulence components are *D*iagonal in that case, i.e. POD modes already correspond to the physical loads. This variant is not really physical but is a simple approach to reveal size effects of the structure. Second, for each considered buffeting loading, either full or diagonal CPSD matrices, modal responses are computed and recombined with the two recombination techniques presented in Sections 3.3.2 and 3.3.3, namely an SRSS/CRSC or a CQC/CCC approach.

Figure 6.4 shows the 2nd and 3rd order moments (in terms of variance/covariance and skewness) of modal loads and modal responses, for the modes 1, 2, and 3, which respond the most to the considered loading. Figures 6.4-(a, b) refer to covariances, while Figures 6.4-(c, d) show third order moments translated in terms of skewness coefficients. Also, 6.4-(a, c) refer to modal loads while 6.4-(b, d) to modal responses. Bars in blue refer to the moments obtained with the full CPSD matrix of turbulence components, while orange bars correspond to the

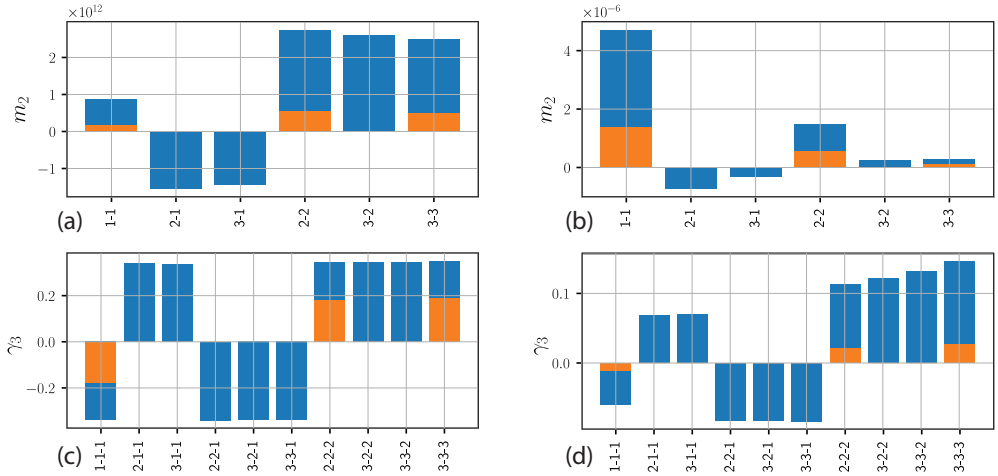


Figure 6.4: 2nd and 3rd order moments of modal forces and modal responses: (a-c) modal loads, (b-d) modal responses. Colors refer to the approximation in spatial coherence: blue=full CPSD matrix of turbulence components, orange=diagonal CPSD matrix of turbulence components (variant D).

diagonal version (variant D). In the latter case, only diagonal elements of the covariance matrix and of the triple correlation tensor are computed since only the SRSS/CRSC combination is considered in that case.

It is observed that the diagonal elements of the covariance matrix and of the triple correlation tensor of modal loads are much smaller in variant D than in the original problem with the actual spatial coherence. At second order, this translates the fact that the many aerodynamic forces distributed along the bridge deck possess a certain coherence in space and little compensations are observed during projection in the modal basis. However, for uncorrelated wind loads at the different nodes of the model, such compensations dominate and the projected wind field is consequently much smaller. At third order, the same reasoning holds and explains why the third statistical moment also drops in variant D. In fact, it drops more proportionally than the second moment since the skewness coefficient (see Equation (2.12), $\gamma_3 = m_3/m_2^{3/2}$) also drops in the same variant. This is explained by the central limit theorem stating that a linear combination of independent processes tends to become Gaussian as the number of independent processes tends to infinity. In variant D, the applied loads are fully independent (uncorrelated), while there exists some partial correlation in the original problem. It is therefore expected that the non-Gaussianity of the combination (i.e. of the modal load) is smaller in variant D.

Figure 6.4 also shows that the skewness of modal responses are smaller than the skewness of modal loads. This is also a consequence of the central limit

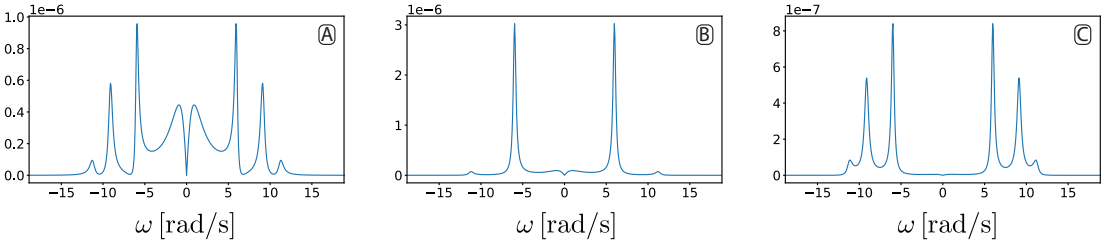


Figure 6.5: Response PSDs, Ff-, at selected nodes, (A) node 10; (B) node 31; (C) node 52.

theorem. Indeed, if the response was quasi-static (fully background), the modal response at a given time would be dependent on the modal load at the same time and there would be the same non-Gaussianity, i.e. the same skewness, in the response and in the loading. As the dynamic (resonant) part of the response becomes more important, for instance in case of small damping, the response at a given time is obtained as a convolution of loads applied at many more instants in the past. This means that there are more terms affecting the response at a given time and the response tends to be more Gaussian, i.e. symmetric in distribution. In the limit case of a fully resonant response, the skewness tends to zero and the response process is statistically symmetric.

Examples of power spectral densities of the structural displacements are three selection points of the bridge are shown in Figure 6.5. The location of points A, B and C correspond to the midspans, see Figure 6.3-a. While the response at node A exhibits a significant background contribution, the responses at nodes B and C are mostly resonant. Similarly, the bispectrum of nodal displacements at nodes A, B and C are reported in Figure 6.6, showing a significant background for node A and resonant responses in modes 1 and 3 for nodes B and C. These power spectral densities and bispectra are obtained with the full CPSD matrix of wind turbulence and with the complete recombination of modal responses. On the other hand, Figure 6.7 shows the same nodal power spectral densities as in Figure 6.5, for the simplified variant "D-". It shows two important points: (i) energy in the response is lost with respect to the most general and complete variant "Ff-", coming from a loss of such energy at the level of modal loads; (ii) the compensation problem discussed above, which results in the flattening of the background (quasi-static) peak around the origin $\omega = 0$, particularly noticeable for the left-most midspan point (A).

In order to evaluate the influence of the modal correlation, two variants of the case with the full CPSM matrix of wind turbulences are considered : in variant "Ff-" modal recombinations are performed with the CQC and CCC (Ff standing for Full-full), while in variant "Fd-" the modal correlations are neglected both at second and third orders, which corresponds to the SRSS and CRSC combination

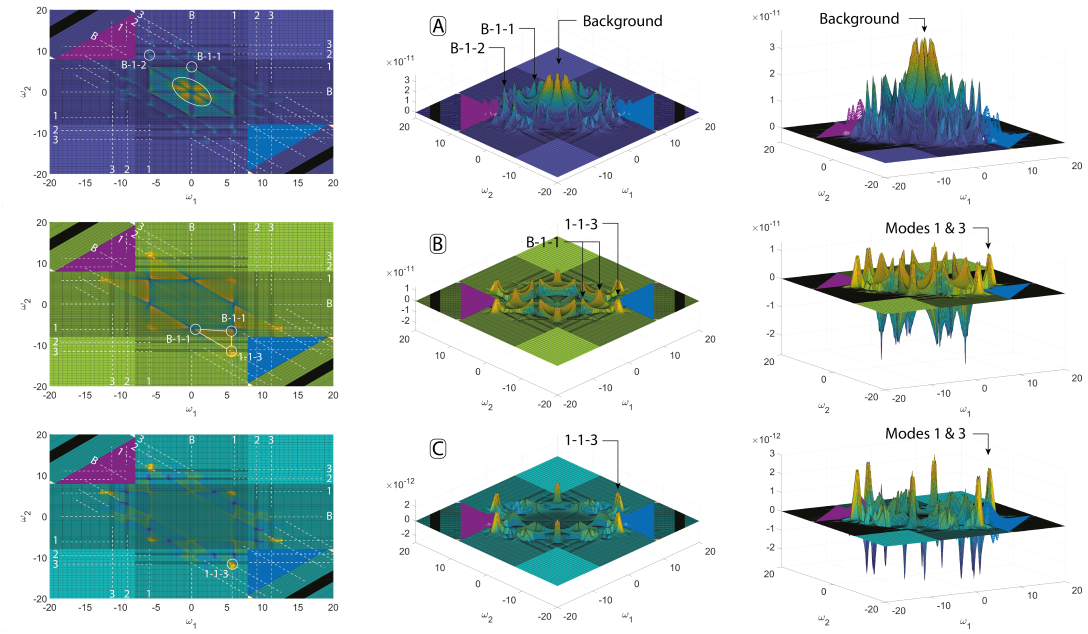


Figure 6.6: Response bispectrum, Ff-, at selected nodes, (A) node 10; (B) node 31; (C) node 52.

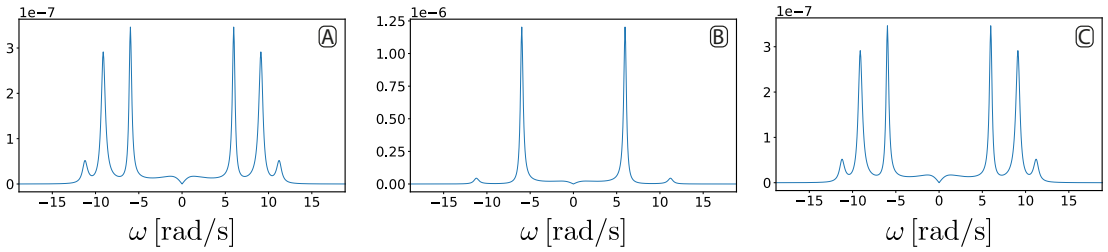


Figure 6.7: Response PSDs, variant D-, at selected nodes, (A) node 10; (B) node 31; (C) node 52.

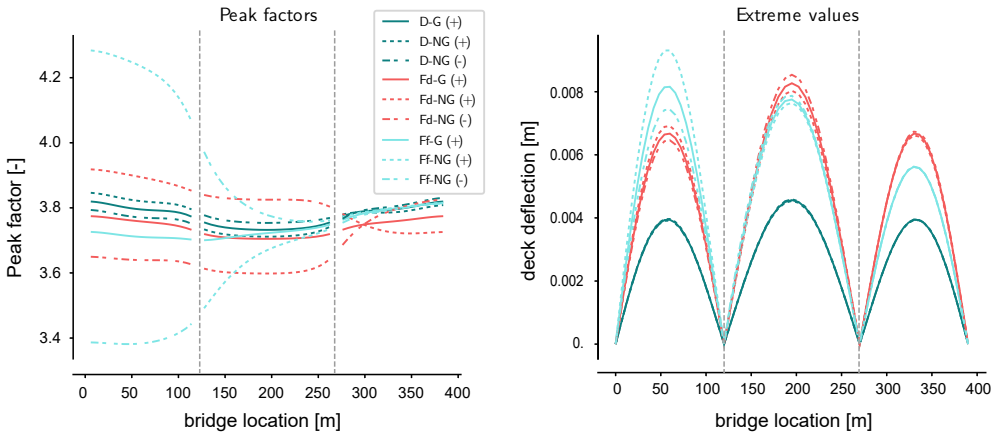


Figure 6.8: Peak factors and fluctuating part of the extreme values of vertical structural displacements. Variants of the structural analysis are: D- (no coherence in wind loads, correlation of modal responses neglected), Ff- (coherent wind loading + complete combinations) and Fd- (coherent wind loading + SRSS/CRSC combinations). For each variant, the extreme values are computed with a Gaussian approach (-G), or with a non-Gaussian approach (-NG+ and -NG−).

methods (in that case, Fd stands for Full-diagonal).

Finally, Figure 6.8 shows the peak factors as well as fluctuating parts of the extreme values of the structural displacements in its transverse direction. This Figure will serve for the discussion on the different modeling assumptions. It shows three categories of curves, each one corresponding to a different variant of modeling (D-, Fd- and Ff-). Then, for each variant, two different way of computing the extreme values are shown : either with a Gaussian formulation (based on Rice's formula), either a non-Gaussian formulation (based on Kareem-Zhao formula [30]), as explained in Sections 3.3.2 and 3.3.3. In the case of a non-Gaussian peak factor, the upper and lower values might differ, so both the positive and negative extreme values are reported. The extreme values represented in Figure 6.8-b correspond to the peak factor multiplied by the standard deviation for each node. To this value, the average displacement field should be added but it is omitted here to ease the reading.

This illustration reveals several important points:

- the complete cubic combination is of primary importance in the determination of third statistical moments. Indeed, the maximum displacement in the left-most span in the two scenarios Ff-NG and Fd-NG results in a difference of about 26% (maximum vertical displacement of 0.0094m vs. 0.007m respectively). This is a consequence of the fact that there are proportionally more elements in the third order moment tensor with respect to the number

of elements on the diagonal, than the same ratio for the covariance matrix. In other words, there are many more elements outside the main diagonal of the third order moment tensor so that, even if the are relatively small, they contribute to the third moment of structural responses.

- the simplified variant "D-" suffers a difference of more than 50% with respect to the complete variant "Ff-", and between 40-45% with the variant "Fd-" in the maximum vertical displacement in the left-most span, which maximum is of 0.004m.

- the comparison of Ff-G and Fd-G shows a small difference in terms of peak factors but well in terms of extreme values (0.0082m vs. 0.0065m, i.e. a difference of 20%). This indicates that the SRSS and CQC combinations provide slightly different standard deviations.

- Figure 6.8 confirms that the loading case of variant D- results in a nearly Gaussian process. This is observable since all three peak factors have very similar values and all three eigen values are virtually superimposed. This is explained by the central limit theorem as stated above.

- the analysis case that results in the largest peak factors is "Ff-", with a significant difference between the positive and the negative peak factors, which indeed translates into significantly different envelopes of extreme values (17% difference in absolute values).

- additionally, for scenarios Ff-, the responses in the central and right-most spans are also nearly Gaussian. This has already been underlined based on the power spectral densities and bispectra of the displacements in the mid-spans of those two spans. It is attributable to the fact that the response is mostly background in the left span, while it is mostly resonant in the two rightmost ones. This effect is slightly less remarked for the Fd- ones, since in such case the use SRSS+CRSC modal recombinations (instead of the CQC/CCC for Ff-, see Sections 3.3.2 and 3.3.3) is not sufficient for a full compensation of effects.

All in all, this example shows the importance of a non-Gaussian (bispectral) analysis in such a case where part of the structure responds in a quasi-static manner. It also highlights the importance of a complete cubic recombination of modal responses as soon as higher order moments are estimated.

In Section 5.1, the two possible extreme (and undesired) scenarios have been presented regarding the proposed methodology; (i) fast but inaccurate, or (ii) accurate but extremely slow. From the mathematical point of view, this is directly related to the kept number of wind modes N_p (and equivalently N_q) when appliying the POD to the incoming wind field, as of Equation (4.53). Indeed,

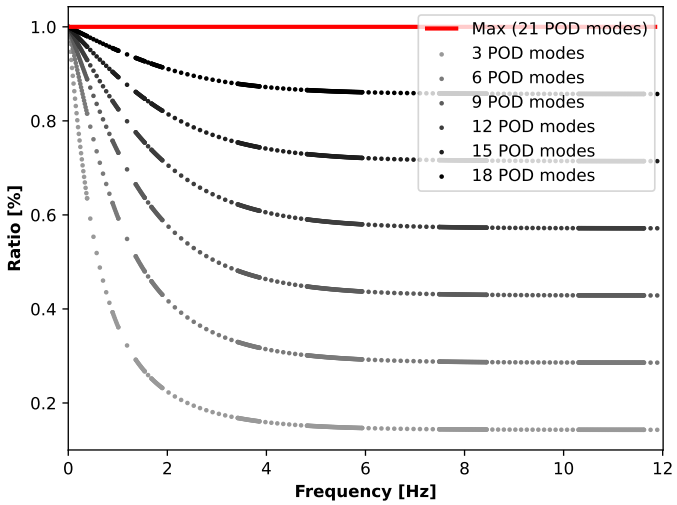


Figure 6.9: Evolution of relative energy with frequency, with changing number of kept POD modes.

use of dimensionality reduction techniques such as POD is of crucial interest when a given phenomenon can be accurately represented by means of a reduced number N_p of independent elements. However, in the results presented so far, all wind modes were considered, that is, $N_p \equiv \texttt{NN}$. From a mathematical point of view, this is indeed equivalent to no POD application at all, since considering all wind modes does lead to no information loss. Nevertheless, even without any modal truncation, the computational cost decreases by a factor of order 10x, which becomes more and more significant as soon as the absolute time (e.g. problem size) increases (i.e. 10days vs. 1day). This speedup is indeed justified by the theoretical concepts that have been detailed in Section 4.8.2, which under a computational point of view, results in the possibility of precomputing some terms instead of having to compute them at each loop iteration. This alone has considerably contributed to prove how POD application in such context is very powerful. Specifically, for the problem considered here, CPU time has varied from about 6 minutes for $N_p = 1$, to 15 minutes for $N_p = 21$. The computational speedup does not increase linearly with the number of POD modes (i.e. around 9 minutes for the case with 9 POD modes). However, this saving might become remarkable for much larger problems.

In practice, there's no unique optimal value for the kept number of wind modes N_p. It is indeed case specific. Retaining between 80-90% of the wind energy might offer a good compromise between speed and accuracy. These ratios also seem to be current practice in other fields of application of POD.

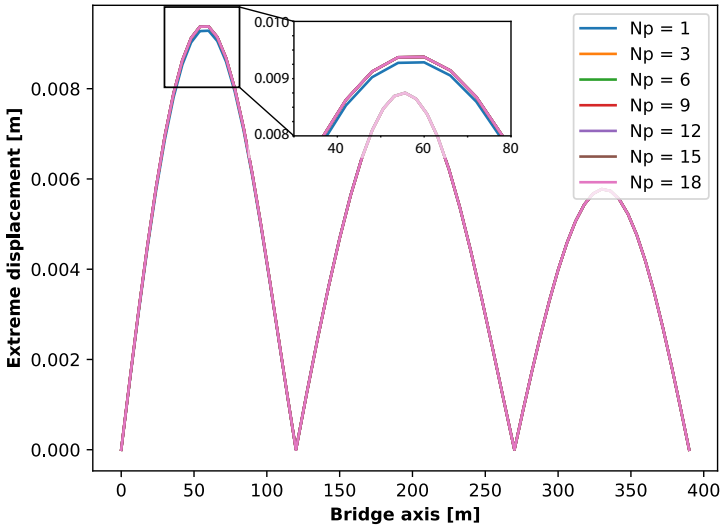


Figure 6.10: Influence of the number of POD modes used in the analysis, on the maximum structural displacements. All curves superimpose for $N_p \geq 3$.

Figure 6.9 shows the ratio of energy contained in the trace of the matrix of eigenvalues of kept modes. Clearly, for the considered case, as soon as the frequency of the load increases, a higher number of POD modes is necessary in order to guarantee the same amount of energy with respect to lower frequencies. This is indeed justified by the fact that, being large scale turbulence and slow-varying process, the main frequency content is towards the lower spectrum of frequencies. As a consequence, in this range, a small number of POD modes already provide more than 70-80% of the total energy. This trend soon vanishes as soon as this evaluation is done at higher frequencies, where in this case, all modes tend to equally contribute to the representation of the loading field.

However, this concerns the wind loading only. Including the structure and investigating the load-structure interaction, one might find that actually, even less POD modes (or in some cases, not the most energetic ones) are the ones actually contributing to the final structural response. This is discussed in details in [81]. Figure 6.10 shows in fact how, for the studied example, already the first POD mode is almost entirely sufficient for accurately reconstructing the full structural response. From 3 to 18 POD modes, the resulting response does not change, meaning that 3 POD modes are the actual only POD modes sufficient to reconstruct the full structural response. Unfortunately however, this information is hardly known beforehand, leading the practitioner to a choice which would, and should, guarantee an adequate precision without having the need of performing the reference case (which would indeed invalidate all this reasoning).

The examples that will follow concern two projects from Bureau d'Étude Greisch[1], for which they provided their models to be used in the context of this Work. The Author is thankful to Them for their kindness and availability.

6.2 Tennet: a transmission line pylon example

The first application concerns the highest among all the steel towers being part of a project of electric network extension via a new 400km line running through Netherlands and Germany, named Wintrack II [140]. It is important to remark that the original design was made for a competition, even before the first pre-design stage. Finally, the project was not selected, so that further, more detailed and accurate studies were not carried by the design office Bureau d'Étude Greisch.

Figure 6.11 shows three representations of the structure: the leftmost is a rendering showing the proposed real setup of the towers installation; the middle is a drawing with relative dimensions in a front view; the rightmost shows the beam model, done using their Finite Element software, FINELG. The structure dimensions are: 71.2m of height; 2.5m base diameter, with 30mm thickness; 0.5m diameter at the top, with 8mm thickness. The main structural material is steel S235.

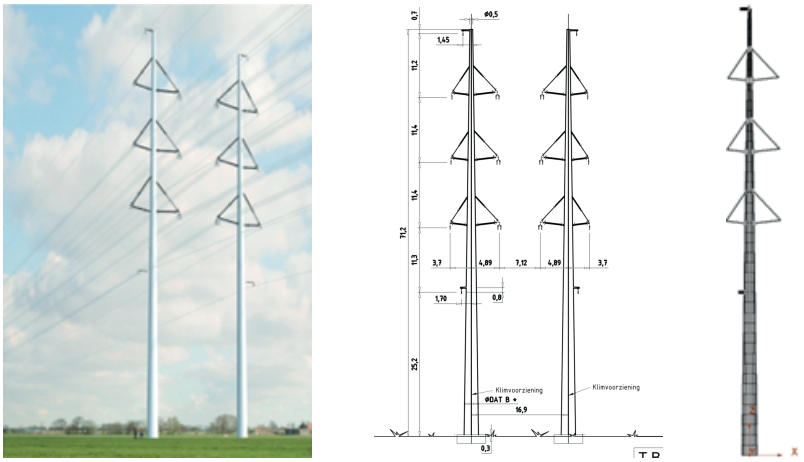


Figure 6.11: (a) Render of the two adjacent towers along the electric line; (b) sketch of the two towers in a front view; (c) view in the X-Z plan of the finite element model of the highest tower Tennet-W4S450 (71.2 m) [140].

A modal analysis was conducted on the finite element model. Some values are reported in Table 6.3 for the first 4 modes of vibration, which for the shape

[1] https://www.greisch.com/en/home/

of the considered structure, are (almost) symmetric along X and Y axes. A damping ratio of 0.3% for all vibration modes has been assumed. In a first global analysis, suitable for very early stages in the process of structural design, the contribution to the total structural damping coming from the transmission cables was not considered. If further, more detailed studies would have followed, this was indeed a contribution that would have been accounted as well. Mode shapes are normalized to a maximum unit displacement.

Table 6.3: Modal structural data for its first 4 modes of vibrations.

Mode	Freq [Hz]	Modal mass [kg]	ξ [%]
1	0.8453	3302.7	0.3
2	0.8456	3315.4	0.3
3	2.3404	1547.1	0.3
4	2.3443	1572.9	0.3

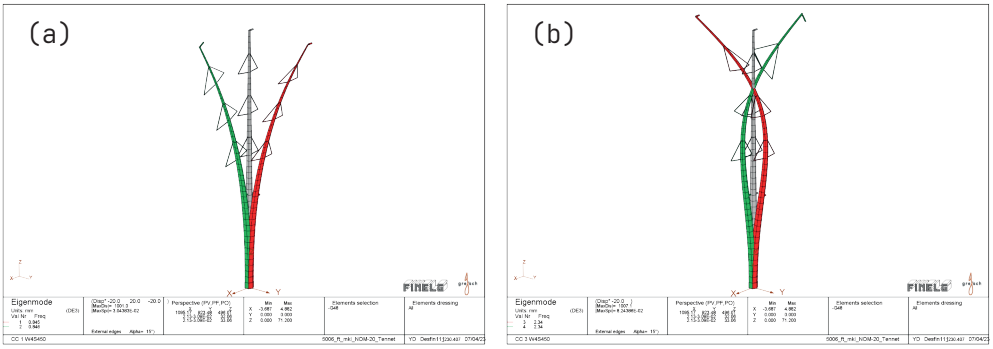


Figure 6.12: Mode shapes: (a) mode 1 (red), mode 2 (green); (b) mode 3 (green), mode 4 (red) [140].

Regarding wind turbulence data, since on-site measurements were not available, Eurocode recommended values were considered. Interested readers can refer to [48] for detailed information. Table 6.4 shows detailed values of some quantities of importance in such an approach.

Table 6.4: Wind turbulence data used for Eurocode approach.

v_{ref} [m/s]	c_{prob} [-]	k_r [-]	α
27	1.12	0.21	0.59

A terrain roughness of category II with $z_0 = 0.2$m and $z_{min} = 4$m were adopted.

A common coherence coefficient $C = 10$ has been considered for all turbulence components, in all spatial directions.

Table 6.5: Standard deviation and wind scale values for spatial wind turbulent components.

	$u(t)$	$v(t)$	$w(t)$
σ [m/s]	6.34	4.76	3.17
L_x [m]	120	30	10
L_y [m]	40	30	7.5
L_z [m]	30	7.5	7.5

Original studies

Originally, the study was carried for comparing two types of wind analyses:

- Equivalent Static wind Loads approach from EN 1991-1-4 [48]. It corresponds to the results under mean wind speed multiplied by a factor $(1 + 7I_u)C_sC_d \geq 1$ which accounts for the wind turbulent component. These values are computed at the pylon reference height $z_{ref} = 0.6z_{max} = 0.6 \cdot 71.2$m. With a turbulence intensity of $I_u = 0.186$ (i.e. 18.6%), $C_sC_d = 0.9$, the Eurocode extreme values correspond to the mean wind speed values multiplied by 2.07.
- A turbulent wind analysis by means of a spectral method, as of Equations (3.39) and (3.40).

Figure 6.13 shows the drag force coefficients, mean wind speeds, and the resulting equivalent wind forces along the tower's elevation.

Comparison with Bispectral Analysis

In this application, three different approaches were adopted:

1. Eurocode equivalent static loads;
2. Spectral analysis;
3. Bispectral analysis, as an extension of the spectral approach for non-Gaussian winds.

As stated in previous sections, the first two were already object of study and comparison of the design office. The third one is the new one proposed in this Work.

In the following illustrations, some labels will be used in order to uniquely identify results from each one of the approaches explored in this application. For clarity, they are:

EC : Eurocode approach, equivalent static load method [48];

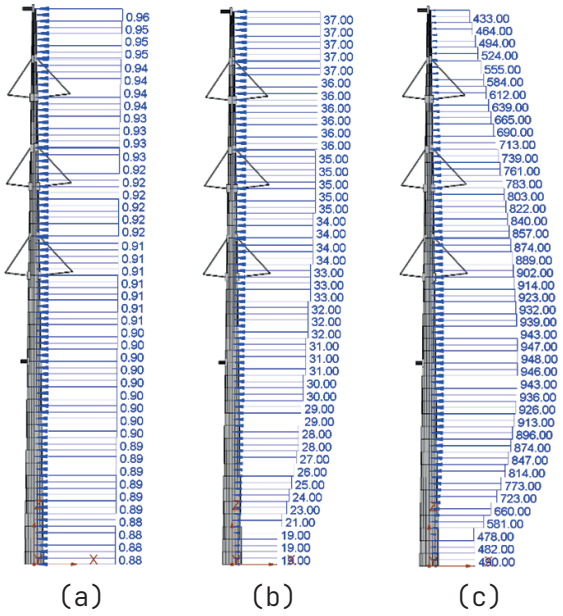


Figure 6.13: Eurocode equivalent static load approach. Variation along the tower height of: (a) C_f wind force drag coefficient; (b) V_m [m/s] mean wind speed; (c) F_b [N/m] mean wind force (per unit length) [140].

ST : static mean contribution of the wind action, obtained through a static analysis (see Section 3.1);

TU : turbulent (fluctuating) contribution, in both Gaussian and non-Gaussian context (TU_{ng}).

Indeed, combining ST with TU results in the extreme values of the response due to, Gaussian or non-Gaussian, wind action (see Equation (2.65)):

$$x_{max} = \overline{x} + g\ \sigma_x \tag{6.6}$$

where $\overline{x}$ is the average response value (ST label), g the Gaussian (or non-Gaussian) peak factor (see Equations (2.66) and (2.69), obtained from γ_3, as explained in Section 2.3.5), σ_x its standard deviation [30] (i.e. label TU refers, in general, to $g\sigma_x$).

Figures 6.14 and 6.15 show the PSD and bispectrum of the nodal displacements of (i) a point at around the mid height of the pylon, and (ii) its top. Indeed, the amplitude (in absolute values) of both the spectrum and bispectrum are greater for the top node. This is the consequence of the fact that the structure behaves as cantilever-beam-like, for which the first vibration mode (in the along-wind direction) happens to be the most significant one. Nevertheless, second mode of vibration also show some small contribution to the overall structural

response. This effect is much clearer for power spectrum, while at third order, this contribution almost flattens out.

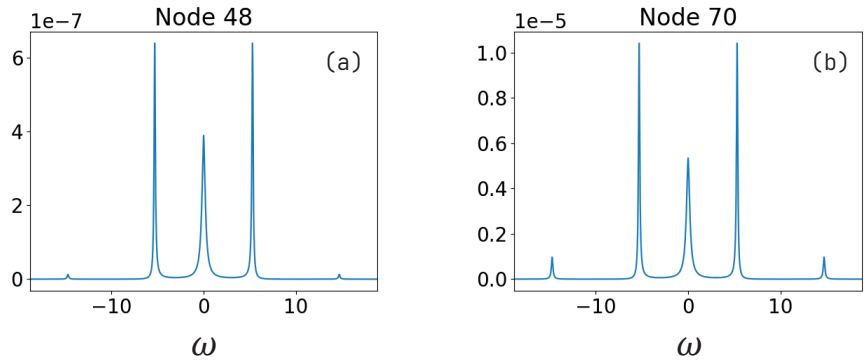


Figure 6.14: PSDs of the displacement along X: (a) node around the mid pylon height; (b) top node.

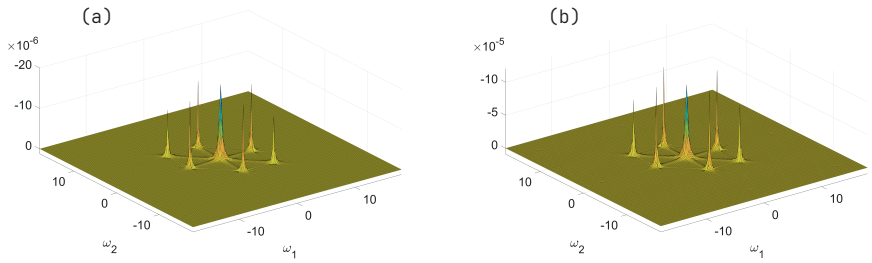


Figure 6.15: Bispectrum of the displacement along X: (a) node around the mid pylon height; (b) top node.

Figures 6.16 and 6.17 show the extreme values of bending moments and shear forces, along the pylon's height.

Table 6.6 finally compares the bending moment values at the bottom of the mast, computed with the different approaches.

In the specific case of high-voltage pylons, when comparing results, it is found that turbulent wind analysis using the Gaussian model results in approximately 30% higher responses compared to the Eurocode equivalent static load method. This difference in results is unlikely to come from the main assumption made in the Eurocode equivalent approach of first mode modal truncation [48], since as also shown from the spectra (PSD and bispectra) from the more complete and accurate stochastic analyses, response contribution come mainly from first vibration mode, with residual contributions coming from the second mode. Instead, it might be the result of an underestimation of the $c_s c_d$ factor of the Eurocode approach, specially if the structure exhibits a comparable amount of quasi-static

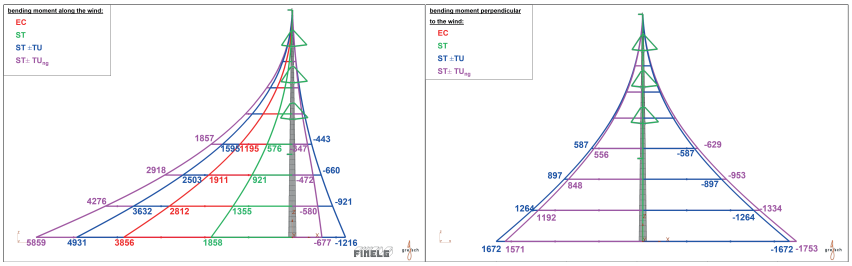


Figure 6.16: Extreme values of bending moments: along-wind (left); across-wind (right). EC = Eurocode approach; ST = Stationary (mean) part; TU = Turbulent part (Gaussian peak factor); TU_{ng} = Turbulent part (non-Gaussian peak factors) [140].

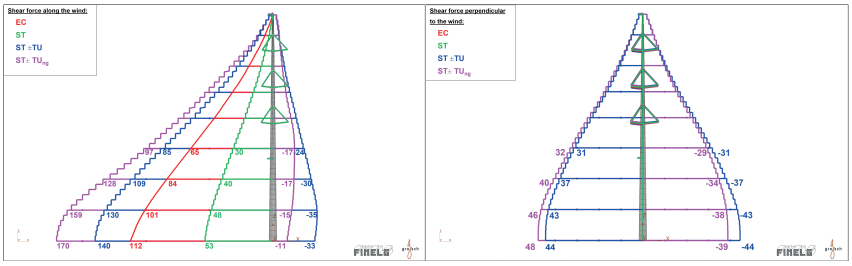


Figure 6.17: Extreme values of shear forces: along-wind (left); across-wind (right). EC = Eurocode approach; ST = Stationary (mean) part; TU = Turbulent part (Gaussian peak factor); TU_{ng} = Turbulent part (non-Gaussian peak factors) [140].

Table 6.6: Resume of numeric results for bending moment and shear force extreme values.

	Bending moment along-wind [kNm]	Bending moment across-wind [kNm]	Shear force along-wind [kN]	Shear force across-wind [kN]
EC	3855.6	0	112.29	0
ST	1857.7	0	53.44	0
ST - TU	-1216.0	-1672.3	-33.11	-44.391
ST + TU	4931.4	1672.3	139.99	44.391
ST - TU_{ng}	-676.5	-1753.1	-11.34	-38.814
ST + TU_{ng}	5858.8	1571.4	170.41	47.751

and dynamic response contributions, that is, a background-to-resonant ratio close to unit value. Furthermore, considering the actual non-Gaussian nature of wind effects increases the responses by an additional 20% (with respect to Gaussian

ones). The extreme values obtained from a bispectral analysis are therefore approximately 50% higher of those estimated via the Eurocode approach. Therefore, in this application, results show how neglecting the non-Gaussian nature of wind loads, maximum (resp. minimum) responses are underestimated (resp. overestimated) in the Gaussian assumption, yielding unsafe (resp. uneconomic) structural design.

6.3 Bridge over the "Grande Ravine"

This case study involves another project conducted by Bureau d'Étude Greisch. The technical and descriptive details here reported are taken from technical reports They kindly provided [141, 142].

It is located in the Municipality of Trois-Bassins, Réunion Island (FR), along the Route of Tamarins (Route des Tamarins) crossing the Valley of the Grande Ravine river. Figure 6.18 shows two views of the area, taken from Google Maps. In Figure 6.18-b, in red it is drawn the exact location of the bridge.

(a) (b)

Figure 6.18: Google Maps views of the site: (a) Macro view; (b) Close view, with red highlights of the exact bridge location crossing the Grande Ravine Valley.

It consists of a 2×2 lanes bridge, officially opened on June 2009, with the aim of reducing the traffic load on the local roads along the west coast of the island. Its main 288m span goes over a 160m deep valley. Figure 6.19 shows an above view of the Bridge, and the Valley running under it.

The deck is made up of orthotropic steel plates, resting on two 20-degree tilted counters, composed of prestressed concrete (see Figure 6.20).

Figures 6.21 and 6.22 show technical drawings of the bridge deck, in a planar and lateral view, respectively.

The structure was designed following a stochastic analysis to turbulent winds, assuming them as Gaussian processes (i.e. spectral analysis). Table 6.7 lists all the relevant turbulence data used to characterise the wind action. Figure 6.23 shows the evolution of the bridge-deck aerodynamic coefficients, with varying

Figure 6.19: Above view of the Bridge over the Grande Ravine river Valley. Courtesy of Bureau d'Étude Greisch.

Figure 6.20: Image of the deck cross-section, during construction stage. Courtesy of Bureau d'Étude Greisch.

incidence angles. For the design, a mean incidence angle $i_0 = -6°$ has been assumed.

Figure 6.24 shows the results obtained in terms of extreme values, comparing a non-Gaussian stochastic analysis (green-positive, blue-negative) with a Gaussian one (red color, original studies). Figure 6.24-a shows the vertical displacements along the bridge deck. Figure 6.24-b the bending moment M_y. Figure 6.24-c the vertical support reactions. Figure 6.24-d the base reaction moments around the Y-axis.

The difference between non-Gaussian and Gaussian results is indeed small. Being both the structure as well as the loading configuration symmetrical, any possible non-Gaussian feature of the wind loading is smoothened out, causing little effects on the overall structural behaviour, with respect to a Gaussian assumption. Nonetheless, small differences can be clearly observed at midspan, where the structure clearly vibrates the most. The highest vertical displacement with a Gaussian assumption is computed at 0.132m, while assuming a non-Gaussian load it increases to 0.139m, with almost a 6% increase (considering the positive formulation of the peak factor). On the other hand, looking at the bending moment across the bridge deck, the positive non-Gaussian formulation shows an overall reduction of the effort at the lateral supports of the main span (while shows an increase of bending moment in the center of the span), while is the negative formulation provides inverted trends, with higher values at the span supports (58200 kNm vs. 55400 kNm at the intermediate supports of the central span) with an increase of 5%, while a decrease of effort in the centre of the span. Then non-Gaussian and Gaussian assumptions provide comparable results in terms of bending moment along the lateral spans. Looking at the results obtained for the vertical support reactions, they understandably follow the considerations made for the vertical displacements of the bridge deck. Finally, for the moment reaction about the Y-axis, there is an inversion of magnitude between positive and negative non-Gaussian formulation with respect to the Gaussian one. This effect might be due to the small slope (0.5%) that the bridge has (from left to right, see Figure 6.22).

Considered the complex morphology of the site (curved valley below, which might affect the wind flow direction) where the bridge locates, it would be interesting to test the structure with uneven turbulent loading, to see if asymmetry in the loading could potentially result in more non-Gaussian features of the resulting responses.

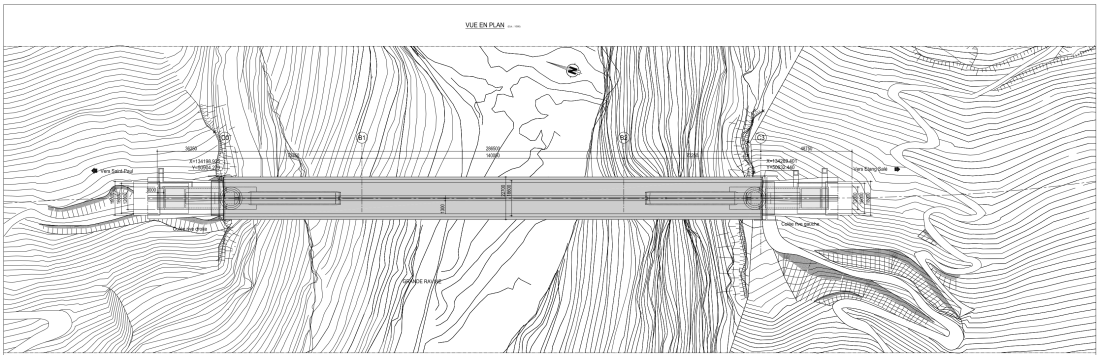


Figure 6.21: In-plane detailed view of the bridge deck. Courtesy of Bureau d'Étude Greisch.

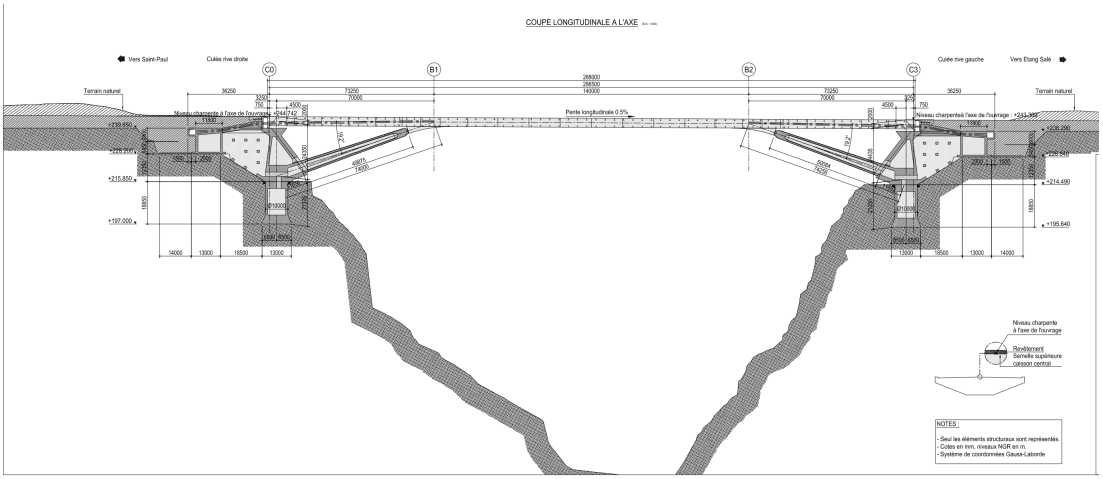


Figure 6.22: Detailed lateral view of the bridge. Courtesy of Bureau d'Étude Greisch.

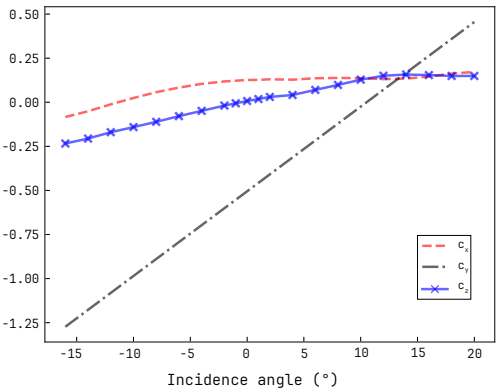


Figure 6.23: Aerodynamic coefficients of the bridge deck.

Table 6.7: Wind turbulence characteristic values.

	$\overline{U}$	$u(t)$	$v(t)$	$w(t)$
μ [m/s]	49.7	0	0	0
σ [m/s]	-	8.4	6.72	5.88
L_x [m]	-	100	50	15
L_y [m]	-	30	30	10
L_z [m]	-	25	20	10
C_x	-	12	12	12
C_y	-	6	6	6
C_z	-	9	9	9

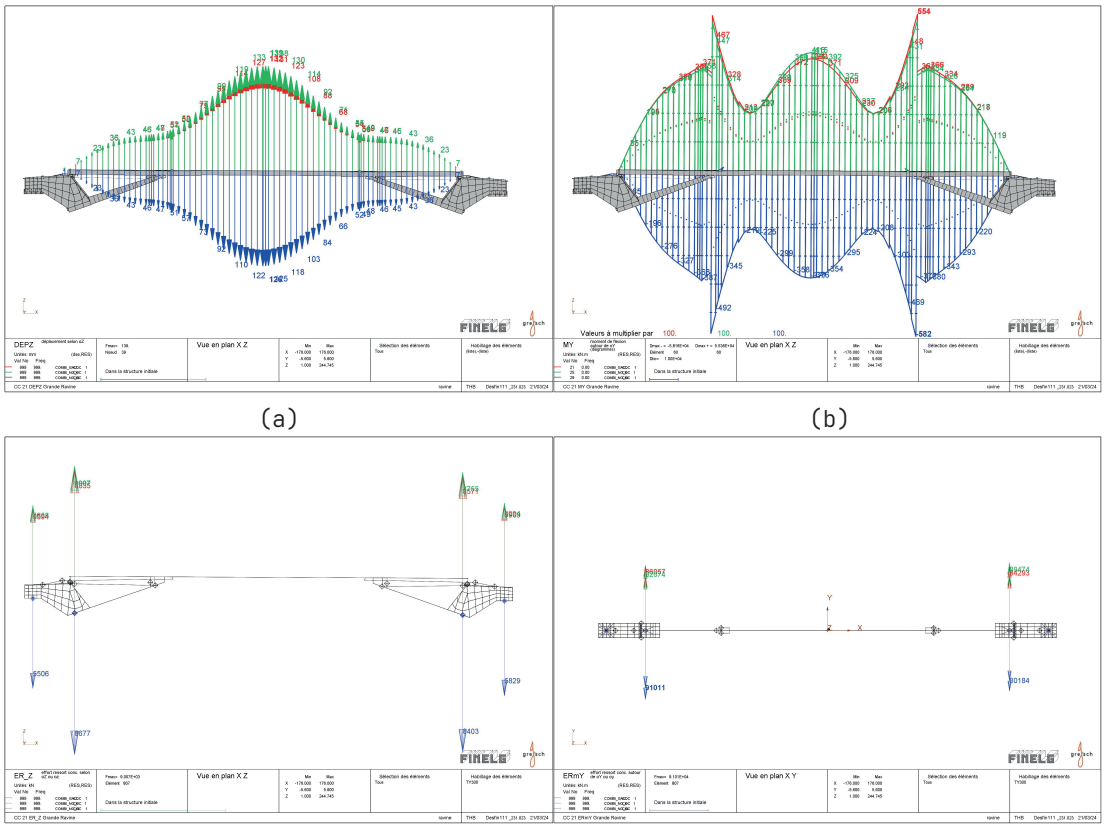


Figure 6.24: Results of a bispectral analysis on the Grande Ravine bridge under non-Gaussian turbulent wind loading. Comparison with previous Gaussian results. Red - original Gaussian results (extreme values). Green - non-Gaussian positive extremes. Blue - non-Gaussian negative extreme. Images generation courtesy of Bureau d'Étude Greisch.

6.4 The SK2BR-curve

In previous Sections applications of Bispectral Analysis to different cases, either academic or real, have been presented and discussed.

They served us to outline how a bispectral analysis might sometimes lead to different results, with respect to current practices, where a Gaussian assumption is made. In some cases, these differences might affect the final design or verification of the structure (as cases similar to the one discussed in Section 6.2). In others, the differences are limited, so that effectively assuming Gaussian wind loads would not harm the final structural design, but it might change some detailing (see Section 6.3).

Also, the example presented in Section 6.1, even though of academic nature, has helped us to show and understand how some simplifications, that are usually made in a Gaussian context (i.e. neglecting the outer-diagonal elements when recombining modal responses statistics), do not have the same mathematical meaning, and so bear considerable different results, at statistical orders higher than the second (i.e. when the Gaussian assumption is dropped).

One should now be able to understand "why" Bispectral analyses are (or might be) important. Yet, all these examples could not give us a clear, general picture of "when" they are actually important, and better not be avoided.

In fact, looking at the problem under an engineering point of view, what would be really appealing is having means to roughly, but quickly, estimate the necessary conditions for which an accurate Bispectral Analysis is strongly recommended, knowing that the final results would be very much different, and when it can be in fact avoided, preferring an easier Spectral (e.g. Gaussian) one (see Section 3.3.2).

Indeed, the source of all these considerations stands in the fact that, a priori, wind loads carry an intrinsic degree of non-Gaussianity, due to reasons that have been extensively discussed in Section 4.4 and Chapter 4. However, what matters the most (in our specific case) is that resulting responses are non-Gaussian as well, so that the interest of using non-Gaussian Bispectral analyses still holds. Nonetheless, as it has been briefly discussed in the introductory part, non-Gaussian nature of wind loads might also be better accounted in those contexts where responses are expected to be Gaussian processes (due to resonance effects, which make them act as undamped harmonic vibrations [14]), but which are highly dependent on the energetic content of the loading process, such as flutter aeroelastic instability phenomena (see Section 4.2).

There are two main factors that influence the resulting non-Gaussian degree of the response of a given structure, undergoing non-Gaussian excitations:

- the degree of non-Gaussianity of the loading itself, e.g. the more non-Gaussian the loading process, the higher the chance that resulting responses

will be non-Gaussian as well;

- the ability of the structure to reduce non-Gaussianity of the loading process through its FRF (Frequency Response Function, see Equation (3.5)) This is directly linked to the frequency contents of both the loading and the structure's main vibration modes. The closer (but not identical) frequency content, the lower the resulting reduction. So that if the loading carries an important degree of non-Gaussianity, the responses will be more likely non-Gaussian as well.

To verify this aspect, several bispectral analyses (for each of the three considered cases, Quetzalapa, Tennet and Grande Ravine) were performed considering different configurations. In this Work, the first source hes been kept fixed, meaning that the wind characteristics and so its non-Gaussian degree were left unchanged. This choice has been made since wind turbulence (and so wind loads) is something intrinsic to the geographical site in which the structure is located, data that is usually collected via on-site measurements. Altering this information would automatically invalidate a fundamental (and physical) aspect of the analysis. Therefore, it has been decided to only let the damping ratio vary, since usually this is a structural parameter that is indeed dependent on the structural system itself, but that can also be modified via external devices (dampers) based on both structural and design requirements (serviceability, comfort criteria [48]).

Table 6.8 resumes all the different damping ratios used, for the three cases.

Table 6.8: Values of damping ratios used, for each case.

	Quetzalapa (6.1)	Tennet (6.2)	Grande Ravine (6.3)
1	0.5%	0.3% [!]	0.3% [!]
2	1%	1%	0.4%
3	2%	1.5%	0.5%
4	3% [!]	2.2%	1%
5	5%	3%	1.5%
6	7.5%	5%	2%
7	10%	8%	2.5%
8	12%	13%	3%
9	15%	20%	5%
10	20%	25%	7.5%
11	25%		10%
12			15%
13			20%

Values marked with [!] are the damping ratio values used in the original studied configuration.

Figure 6.25 shows the resulting curves of skewness to background-resonant ratios. On the abscissa, the background-resonant ratio reads [29]

$$BR = \frac{m_{2,b}}{m_{2,r}} \tag{6.7}$$

where

$$m_{2,b} = \frac{m_{2,p}}{k^{*2}} \tag{6.8}$$

and

$$m_{2,r} = \frac{\pi \omega_0 S_p(\omega_0)}{2\xi k^{*2}} \tag{6.9}$$

are the background (i.e. quasi-static) and resonant components of a given modal response, being $m_{2,p}$ the second order moment (i.e. variance) of the relative modal loading $p(t)$, k^* the modal stiffness, ω_0 the modal natural circular frequency, ξ the modal damping ratio and $S_p(\omega_)$ the PSD of the modal load evaluated at modal frequency ω_0. Physically, the BR ratio quantifies how much the modal response is background (i.e. quasi-static) with respect to its resonant component. Ratios smaller than 1 would indicate that the modal response is mainly resonant, while ratios greater than one would suggest that the dominant structural behaviour is quasi-static.

On the other hand, the skewness ratio on the ordinate refers to the ratio of skewness of modal response over skewness of the relative modal loading. That is:

$$SK = \frac{\gamma_{3,q}}{\gamma_{3,p}} \tag{6.10}$$

where $\gamma_{3,q}$ and $\gamma_{3,p}$ are the skewness of modal response and load respectively.

Figure 6.25 shows the resulting SK2BR curves (for each case, left to right) for the principally contributing modal responses. In all cases, as soon as the background-resonant (BR) ratio increases, the skewness (SK) ratio tends to the limit case in which the SK ratio has unitary value, i.e. the skewness of the modal response $\gamma_{3,q}$ is identical to the skewness of the modal load $\gamma_{3,p}$ (complete conservation). Looking carefully, it might seem that some curves have been cut (purposely) at the intersection with the horizontal asymptote. However, the Author can prove that no cut was made. While it would be possible to have those curves slightly surpass the horizontal asymptote, this is very unlikely to happen (since a linear system has been assumed), and if so, under very special conditions. Therefore, the unitary SK ratio is the ultimate case that one could possibly reach in most cases, which is effectively reached under some (extreme) large damping ratio configurations.

While some might argue that these curves have been drawn assuming some damping conditions that would be extremely rare (if not impossible) to have in a

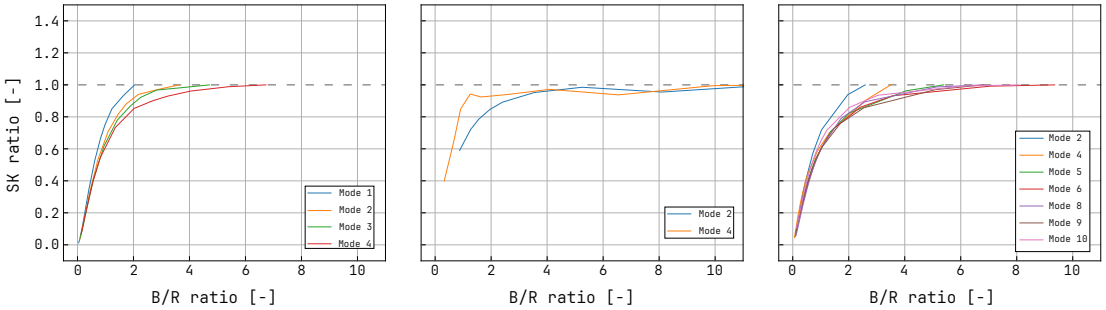


Figure 6.25: Skewness to background-resonant ratios (SK2BR) curves, for the main structural vibration modes.
Quetzalapa (Left); Tennet (Centre); Grande Ravine (Right).

real scenario, it is important to look at these results from a different perspective. Figure 6.26 draws the same results, but grouped into the same, unified, figure. As first observation, it can be noticed how for three different structural systems, not only the trends are almost identical, but also the values are quite close. This is of fundamental importance, proving how these curves are in effect consistent no matter the structural/load characteristics. Moreover, for BR ratios smaller than 1 (i.e. more resonant than background, quasi-static behaviour), the SK ratio can reach values as high as 0.5, which means that the modal responses could carry up to half the total degree of non-Gaussianity of modal loads.

So that, neglecting the non-Gaussian nature of the wind loads could be harmful if the skewness of the wind load (see Equation (6.11) for the analytical formula at the level of nodal wind forces [5]) would be high enough (> 0.2) to result in strong-to-medium non-Gaussian responses.

$$\gamma_{f_i} = 3I_{u_i} \frac{1 + 2\frac{c_{1_i}^2}{4c_{0_i}^2}\frac{I_{w_i}^2}{I_{u_i}^2} + \frac{c_{1_i}^2}{4c_{0_i}^2}\left(1 + \frac{c_{2_i}}{2c_{0_i}}\right)\frac{I_{w_i}^4}{I_{u_i}^4}(1 + I_{u_i}^2)}{\left[1 + \frac{c_{1_i}^2}{4c_{0_i}^2}\frac{I_{w_i}^2}{I_{u_i}^2}\right]^{\frac{3}{2}}} \tag{6.11}$$

in which small terms have been neglected with respect to the formulation given in [5].

The SK2BR-curve in Figure 6.26 formalises an important, novel point of view in regard to buffeting analyses. If the skewness of the wind load in Equation (6.11) (proportional to $3I_u$, where I_u is the turbulence intensity, see Section 4.1) is high enough, then a quick assessment of the background-resonant BR ratio by means of a spectral analysis (second statistical order, see Section 3.3.2) would serve to estimate if a more accurate non-Gaussian bispectral analysis is required for the case studied. In fact, for BR ratios > 0.5, if the wind load contains high non-Gaussian features, then Figure 6.26 tells us that it is in fact recommended to perform a bispectral analysis a since a considerable degree non-Gaussianity is

expected at the level of modal responses, and so therefore structural responses too.

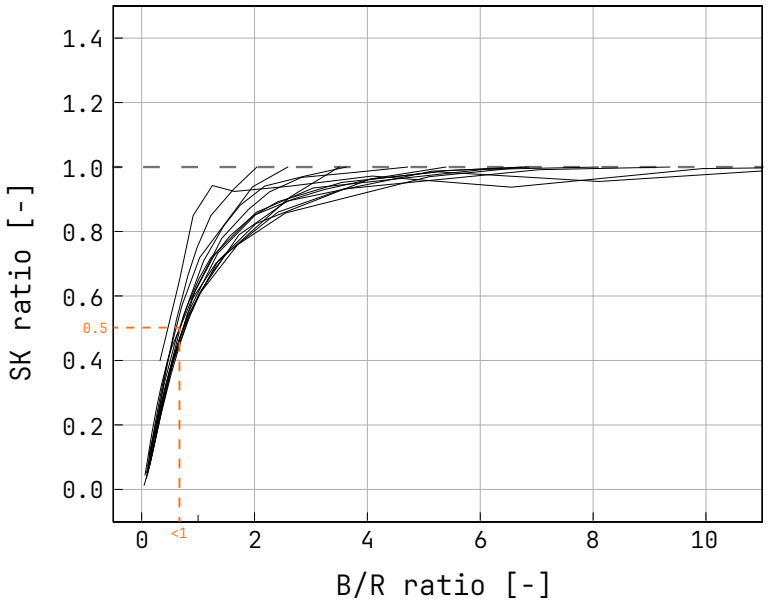


Figure 6.26: Unified SK2BR curve.

Chapter 7

Conclusions & further developments

In this Thesis, the problem of Bispectral Analysis was covered.

Chapters 1 to 3 served at giving a complete introduction and overview of the mathematical aspects involved in a Bispectral Analysis.

In Chapter 4, after having firstly outlined the mathematical reasons for which the wind load carries an intrinsic degree of non-Gaussianity, a generalised wind loading model has been formulated, that given the probabilistic description of the wind turbulence, evaluates the stochastic quantities of interest of the resulting wind load, regardless of the degree of the model of transformation. Then, in the context of this Thesis, its specialisation to a quadratic (second) transformation has been detailed, from which PSDs and Bispectra of wind loads have been derived.

In today's current practice, while more and more authors in the scientific community recognise the non-Gaussian nature of wind loading (from actual measurements), very few authors do effectively account for this aspect of the buffeting load. Also, many considering a time domain approach to the resolution of the general equations of motion (see (3.1)) [72, 7]. In [70], authors have also extended the application of non-Gaussian wind loads to a system with nonlinear mechanical behaviour.

In a spectral approach, currently, the practical design of structures is based on a Gaussian assumption. There is no commercial software nor any research-oriented Finite Element code able to deal with non-Gaussian analysis in the frequency domain. In that, this Work aimed at encouraging the use of stochastic approaches, extending to even more complex scenarios. In particular, the common assumption that is made in practice of Gaussian wind loads is no longer kept valid (i.e. linearisation of buffeting forces), and instead a non-Gaussian formulation is

considered. This has also shown that this is not only doable, with the proposed methodology, but sometimes strongly recommended.

In Section 4.8 Proper Orthogonal Decomposition (POD) was discussed. The origins of such mathematical techniques go back to 19^{th}-20^{th} century. After a very brief historical and mathematical recall of the most important concepts behind such techniques, their leverage in Wind Engineering applications was recalled. In that, Solari[1] in [43, 82] has done a monumental work in collecting all the applications of POD-like techniques within the macro-areas influencing modern Wind Engineering. Following, Carassale et al. in [81] detailed the application of POD techniques to a spectral approach to the buffeting problem, with an approach named "Double Modal Transformation", in which they coupled POD techniques intrinsically used to solve the Eigenvalue problem for the determination of the modal matrix to a POD decomposition of the spatial wind field into a set of principal components. Acknowledging the powerfulness of such an approach, in Section 4.8.2 extension of POD to a non-Gaussian context was formulated, establishing a direct relation between the bispectra of non-Gaussian buffeting loads and the POD modes of the PSDs of the base wind turbulence.

Then, Chapter 5 covered the numerical part, which is an important aspect of this Work. The novel MESHER algorithm has been presented, and its core features discussed. Combining both the theoretical benefits of POD and the power and efficiency of the proposed algorithm, non-Gaussian buffeting analysis of real (possibly large) civil engineering structures has been tackled, for the first time.

Finally, in Chapter 6, a series of examples have been presented as a concrete example of application of bispectral analyses. They covered both academic and real examples. The academic ones, as the Quetzalapa Bridge example presented in Section 6.1 are indeed a useful tool to provide, for anyone interested in reproducing the results presented in this Work. On the other hand, the real examples have served us to show that with the proposed methodology, accurate bispectral analyses are now applicable to relatively large MDOFs structures, with close to a thousand degrees-of-freedom actually loaded by non-Gaussian buffeting loads (i.e. counting much more than a thousand in total). Moreover, in less than 6h of computational time on a single machine, using only 4 cores. This hardware configuration has been deliberately chosen in this first stage of developments as a way to test the applicability of the novel algorithm even on older machines not having all the physical cores that modern boards provide (8, 16 or more).

[1] Named amongst the Fathers of modern Wind Engineering.

There are however several ways in which this Work can (and should!) be improved. Starting from the last point, discussed in detail in Chapter 5, the numerical side, which is indeed a huge part of this Work, has yet lots of improvement space that must be covered, so to enhance even more the computational power of the proposed methodology, and extend feasibility of bispectral analyses on a single machine of even bigger models. Among the most important aspects, the inclusion of distributed memory architectures parallel paradigms (`MPI`) to leverage execution of bispectral analyses on multiple machines (e.g. clusters), whenever available. This is specially important (as per the current implementation) if tackling very big models, with more than tens of thousands degrees-of-freedom which are under the non-Gaussian buffeting action. Currently, these kinds of model would exceed several days of computing time on a single machine. Also, improve and extend GPU support, for both shared and distributed memory systems, in case of single or multi-GPU hardware support. This is potentially what would really provide massive scalability capabilities to the MESHER algorithm.

Another important aspect to further improve the numerical part, is a better integration of the current algorithmic arrangement with a time domain approach. Ideally, its ability to generate an optimal meshing pattern, based on all the aspects discussed in detail in Chapter 5, provided discrete records of wind velocities (or pressures), either measured or simulated [143]. This feature could be based starting from the generation a discrete bispectrum from a time series (through the use of Discrete Fourier Transform algorithms) (as it has been done in the first developments of the FINELG2020 Research Project, for the validations with a Monte Carlo approach [137, 138]), and building an optimised meshing pattern on top of it, reconstructing the required information where missing.

On the mathematical side, while this Work aimed at establishing a general framework for the analysis of structures under non-Gaussian wind loads, it specialised in the Bispectral analysis, therefore limited to third statistical order only. Extension of this framework to even higher-order, to say at least fourth order with explicit formulations of the trispectrum, could finally provide all the necessary tools for better tackling the problem of extreme values computation, discussed in Section 2.3.5. This would also allow the drop of the necessity of an assumed, fixed relation between the skewness γ_3 and the excess kurtosis γ_e coefficients, as it was explained in Section 2.3.5 (see Equation (2.71)). Currently, without it one would not be able to have estimates of non-Gaussian peak factors (see Equations (2.69) and (2.65)), and so interest in Bispectral analyses in the first place. While it is indeed a valid mathematical assumption in most of the cases to which common Wind Engineering applications can be related to [32], having a way to directly estimate fourth order statistics is indeed a preferred option, and something to strive for.

Another way of further generalising the proposed methodology is to include other loading phenomena than the wind buffeting loading, such as wave loading [144, 145, 146], or fatigue [147, 148]. Also, while in this Work only linear systems have been considered, further research could include application of bispectral analyses to cases where the loading is non-Gaussian and at the same time the structure shows some nonlinear behaviour, which might introduce an additional non-Gaussian degree to the resulting structural responses. This would be in continuity of the work already done by other authors in this direction [70].

Last, but not least, the most important aspect, which is at the base of this Work and many other applications in modern Wind Engineering to the general aeroelastic problem (including the buffeting problem and all the aeroelastic unstable phenomena, discussed in the introductory part in Section 4.2): the assumption that the base turbulence (i.e. what here has been referred to as the *elementary* wind turbulent components) is a zero-mean, random, **Gaussian** process. While this assumption has been proved by many authors, for which the full list of references would cover at least half a page, in last the decade few authors ([10] among others) have started claiming that in reality, under some cases, even the base wind turbulence might show some degree of non-Gaussianity, and that the most fundamental assumption of Gaussian turbulence does not hold in all circumstances. In this scenario, a completely new kind of research could investigate the mathematical formulation of the generalised wind model presented in this Work, but even more generalised, accounting for the possibility of the introduction of a completely new concept: the Bispectrum of wind turbulence.

Indeed, from this enhanced generalised model formulation would certainly benefit all those physical applications in which it is already known that the base process is itself already a non-Gaussian random process, far beyond the realm of Wind Engineering.

Bibliography

[1] Emil Simiu and Robert H Scanlan. *Wind effects on structures: fundamentals and applications to design*. Vol. 688. John Wiley New York, 1996.

[2] Zihang Liu et al. "Uncertainty propagation of turbulence parameters for typhoon and Non-typhoon winds in buffeting analysis of Long-span bridges". In: *Engineering Structures* 291 (2023), p. 116491. ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2023.116491. URL: https://www.sciencedirect.com/science/article/pii/S0141029623009069.

[3] Pál Révész. *The laws of large numbers*. Vol. 4. Academic Press, 2014.

[4] Fourth Edition, Athanasios Papoulis, and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. McGraw-Hill Europe: New York, NY, USA, 2002.

[5] Vincent Denoël. "Application des méthodes d'analyse stochastique à l'étude des effets du vent sur les structures du génie civil". PhD dissertation. Université de Liège, 2005.

[6] Ahsan Kareem. "Nonlinear wind velocity term and response of compliant offshore structures". In: *Journal of Engineering Mechanics* 110.10 (1984), pp. 1573–1578.

[7] Claudio Borri and Wolfhard Zahlten. "Fully Simulated Nonlinear Analysis of Large Structures Subjected to Turbulent Artificial Wind*". In: *Mechanics of Structures and Machines* 19.2 (1991), pp. 213–250.

[8] V. Gusella and A.L. Materazzi. "Non-Gaussian along-wind response analysis in time and frequency domains". In: *Engineering Structures* 22.1 (2000), pp. 49–57. ISSN: 0141-0296. DOI: https://doi.org/10.1016/S0141-0296(98)00074-1.

[9] M Gioffrè, V Gusella, and M Grigoriu. "Non-Gaussian Wind Pressure on Prismatic Buildings. I: Stochastic Field". In: *Structural Engineering* 127.9 (2001), pp. 981–989. ISSN: 0733-9445.

[10] Marcelo Chamecki. "Persistence of velocity fluctuations in non-Gaussian turbulence within and above plant canopies". In: *Physics of Fluids* 25 (Nov. 2013), p. 115110. DOI: 10.1063/1.4832955.

[11] Wei Cui, Lin Zhao, and Yaojun Ge. "Non-Gaussian turbulence induced buffeting responses of long-span bridges based on state augmentation method". In: *Engineering Structures* 254 (2022), p. 113774. ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2021.113774. URL: https://www.sciencedirect.com/science/article/pii/S0141029621018502.

[12] Vittorio Gusella and Annibale Luigi Materazzi. "Non-Gaussian Response of MDOF Wind-Exposed Structures: Analysis by Bicorrelation Function and Bispectrum". In: *Meccanica* 33 (1998), pp. 299–307. DOI: 10.1023/A:1004355317308.

[13] A Preumont. *Random Vibration and Spectral Analysis/Vibrations aléatoires et analyse spectral*. Vol. 33. Springer Science & Business Media, 1994.

[14] R. W. Clough and J. Penzien. *Dynamics of Structures*. International student edition. McGraw-Hill, 2003. ISBN: 9780070113923. URL: https://books.google.be/books?id=UdxRAAAAMAAJ.

[15] Edwin T Jaynes. "Prior probabilities". In: *IEEE Transactions on systems science and cybernetics* 4.3 (1968), pp. 227–241.

[16] W.B. Collis, P.R. White, and J.K. Hammond. "HIGHER-ORDER SPECTRA: THE BISPECTRUM AND TRISPECTRUM". In: *Mechanical Systems and Signal Processing* 12.3 (1998), pp. 375–394. ISSN: 0888-3270. DOI: https://doi.org/10.1006/mssp.1997.0145. URL: https://www.sciencedirect.com/science/article/pii/S088832709790145X.

[17] Mario Di Paola. "Digital simulation of wind field velocity". In: *Journal of Wind Engineering and Industrial Aerodynamics* 74-76 (1998), pp. 91–109. ISSN: 0167-6105. DOI: https://doi.org/10.1016/S0167-6105(98)00008-7.

[18] Akira Masuda and Yi-Yu Kuo. "A note on the imaginary part of bispectra". In: *Deep Sea Research Part A. Oceanographic Research Papers* 28.3 (1981), pp. 213–222. ISSN: 0198-0149. DOI: https://doi.org/10.1016/0198-0149(81)90063-7.

[19] Margaux Geuzaine, Michele Esposito Marzino, and Vincent Denoël. "Third-order spectral analysis of an oscillator subjected to wind loads with complex-valued bispectrum". In: *CWE Workshop 2022: Advanced modeling of stochastic Wind Effects and Vibrations*. Presented at the CWE Workshop. 2022.

[20] Shimizu Hisaji and Inoue Takeyasu. "Machine Fault Diagnosis by Vibrational Analysis: Exploratory Introduction of Bispectrum Method". In: 27 (1978). I think it is in Japanese - I couldnt get access to the original document, pp. 51–60.

[21] I. Tawfiq and T. Vinh. "Nonlinear Behaviour of Structures Using the Volterra Series—Signal Processing and Testing Methods". In: *Nonlinear Dynamics* 37 (2004), pp. 129–149. DOI: 10.1023/B:NODY.0000042909.51421.61.

[22] Shen Guoji et al. "Theoretical and experimental analysis of bispectrum of vibration signals for fault diagnosis of gears". In: *Mechanical Systems and Signal Processing* 43.1 (2014), pp. 76–89. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2013.08.023. URL: https://www.sciencedirect.com/science/article/pii/S0888327013004342.

[23] Martin Schetzen. "Nonlinear system modelling and analysis from the Volterra and Wiener perspective". In: *Block-oriented Nonlinear System Identification*. Springer, 2010, pp. 13–24.

[24] Alexandre Depouhon, Emmanuel Detournay, and Vincent Denoël. "Accuracy of one-step integration schemes for damped/forced linear structural dynamics". In: *International Journal for Numerical Methods in Engineering* 99.5 (2014), pp. 333–353.

[25] Masanobu Shinozuka and George Deodatis. "Simulation of stochastic processes by spectral representation". In: (1991).

[26] Vincent Denoël and Hervé Degée. "Influence of the non-linearity of the aerodynamic coefficients on the skewness of the buffeting drag force". In: *Wind and Structures* 9.6 (2006).

[27] "Multiple timescale spectral analysis". In: *Probabilistic Engineering Mechanics* 39 (2015), pp. 69–86.

[28] A. G. Davenport. "The application of statistical concepts to the wind loading of structures." In: vol. 19. 4. 1961, pp. 449–472. DOI: 10.1680/iicep.1961.11304.

[29] Vincent Denoël. "On the background and biresonant components of the random response of single degree-of-freedom systems under non-Gaussian random loading". In: *Engineering Structures* 33.8 (2011), pp. 2271–2283. ISSN: 0141-0296. DOI: https://doi.org/10.1016/j.engstruct.2011.04.003.

[30] Dae Kun Kwon and Ahsan Kareem. "Peak Factor for Non-Gaussian Load Effects Revisited". In: *Journal of Structural Engineering* 137 (Dec. 2011), pp. 1611–1619. DOI: 10.1061/(ASCE)ST.1943-541X.0000412.

[31] Alan G. Davenport. "NOTE ON THE DISTRIBUTION OF THE LARGEST VALUE OF A RANDOM FUNCTION WITH APPLICATION TO GUST LOADING." In: 1964. URL: https://api.semanticscholar.org/CorpusID:109127335.

[32] François Rigo, Thomas Andrianne, and Vincent Denoël. "A de-mixing approach for the management of large negative peaks in wind tunnel data". In: *Journal of Wind Engineering and Industrial Aerodynamics* 206 (2020), p. 104279. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2020.104279. URL: https://www.sciencedirect.com/science/article/pii/S0167610520301896.

[33] Vincent Denoël, Michele Esposito Marzino, and Margaux Geuzaine. "A multiple timescale approach of bispectral correlation". In: *Journal of Wind Engineering and Industrial Aerodynamics* 232 (2023), p. 105282.

[34] K.H. Huebner et al. *The Finite Element Method for Engineers*. A Wiley-Interscience publication. Wiley, 2001. ISBN: 9780471370789.

[35] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering. Dover Publications, 2012. ISBN: 9780486135021.

[36] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Science, 2005. ISBN: 9780080472775.

[37] T. Canor, N. Blaise, and V. Denoël. "Efficient uncoupled stochastic analysis with non-proportional damping". In: *Journal of Sound and Vibration* 331.24 (2012), pp. 5283–5291. ISSN: 0022-460X. DOI: https://doi.org/10.1016/j.jsv.2012.07.019. URL: https://www.sciencedirect.com/science/article/pii/S0022460X12005457.

[38] Ray W Clough and Soheil Mojtahedi. "Earthquake response analysis considering non-proportional damping". In: *Earthquake Engineering & Structural Dynamics* 4.5 (1976), pp. 489–496.

[39] Erik H Vanmarcke. "Structural response to earthquakes". In: *Developments in geotechnical engineering*. Vol. 15. Elsevier, 1976, pp. 287–337.

[40] Armen Der Kiureghian and Yutaka Nakamura. "CQC modal combination rule for high-frequency modes". In: *Earthquake engineering & structural dynamics* 22.11 (1993), pp. 943–956.

[41] Michele Esposito Marzino and Vincent Denoël. "Non-Gaussian buffeting analysis of large structures by means of a Proper Orthogonal Decomposition". In: *Journal of Wind Engineering and Industrial Aerodynamics* 242 (2023), p. 105576. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2023.105576.

[42] Y. Tamura and A. Kareem. *Advanced Structural Wind Engineering*. Springer Japan, 2013. ISBN: 9784431543374.

[43] Giovanni Solari, Luigi Carassale, and Federica Tubino. "Proper orthogonal decomposition in wind engineering - Part 1: A state-of-the-art and some prospects". In: *Wind and Structures* 10 (2007), pp. 153–176.

[44] John David Anderson and John Wendt. *Computational fluid dynamics*. Vol. 206. Springer, 1995.

[45] John David Anderson et al. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2013.

[46] Bert Blocken. "50 years of computational wind engineering: past, present and future". In: *Journal of Wind Engineering and Industrial Aerodynamics* 129 (2014), pp. 69–102.

[47] Theodore Stathopoulos. "Computational wind engineering: Past achievements and future challenges". In: *Journal of Wind Engineering and Industrial Aerodynamics* 67 (1997), pp. 509–532.

[48] *EN 1991-1-4:2005+A1 Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions*. EN. Brussels: CEN, 2009.

[49] Yin Zhou et al. "Along-Wind Load Effects on Tall Buildings: Comparative Study of Major International Codes and Standards". In: *Journal of Structural Engineering-asce - J STRUCT ENG-ASCE* 128 (June 2002). DOI: 10.1061/(ASCE)0733-9445(2002)128:6(788).

[50] A. G. Davenport. "The response of slender, line-like structures to a gusty wind." In: vol. 23. 3. 1962, pp. 389–408. DOI: 10.1680/iicep.1962.10876.

[51] R.H. Scanlan. "The action of flexible bridges under wind, II: Buffeting theory". In: *Journal of Sound and Vibration* 60.2 (1978), pp. 201–211. ISSN: 0022-460X. DOI: https://doi.org/10.1016/S0022-460X(78)80029-7.

[52] Etienne Cheynet et al. "Application of short-range dual-Doppler lidars to evaluate the coherence of turbulence". In: *Experiments in Fluids* 57 (Nov. 2016), pp. 1–17. DOI: 10.1007/s00348-016-2275-9.

[53] Y.C. Fung. *An Introduction to the Theory of Aeroelasticity*. Dover Publications, 1993. ISBN: 9780486469362.

[54] Grigorios Dimitriadis. *Introduction to nonlinear aeroelasticity*. John Wiley & Sons, 2017.

[55] Gianni Bartoli and Claudio Mannini. "A simplified approach to bridge deck flutter". In: *Journal of Wind Engineering and Industrial Aerodynamics* 96.2 (2008), pp. 229–256. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2007.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0167610507001183.

[56] G. Bartoli and M. Righi. "Flutter mechanism for rectangular prisms in smooth and turbulent flow". In: *Journal of Wind Engineering and Industrial Aerodynamics* 94.5 (2006). The eighth Italian National Conference on Wind Engineering IN-VENTO-2004, pp. 275–291. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2006.01.014. URL: https://www.sciencedirect.com/science/article/pii/S0167610506000043.

[57] Luca Caracoglia. "Simulation of linear and non-linear propagation effects of a random turbulence field on bridge flutter instability". In: *Journal of Wind Engineering and Industrial Aerodynamics* 99.9 (2011). 9th UK Conference on Wind Engineering (September, 2010), pp. 945–954. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2011.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0167610511001152.

[58] Tajammal Abbas, Igor Kavrakov, and Guido Morgenthal. "Methods for flutter stability analysis of long-span bridges: a review". In: *Proceedings of the Institution of Civil Engineers-Bridge Engineering*. Vol. 170. 4. Thomas Telford Ltd. 2017, pp. 271–310.

[59] Charles HK Williamson and R Govardhan. "Vortex-induced vibrations". In: *Annu. Rev. Fluid Mech.* 36 (2004), pp. 413–455.

[60] R.H. Scanlan. "The action of flexible bridges under wind, I: Flutter theory". In: *Journal of Sound and Vibration* 60.2 (1978), pp. 187–199. ISSN: 0022-460X. DOI: https://doi.org/10.1016/S0022-460X(78)80028-5.

[61] E. Strømmen and E. Hjorth-Hansen. "The buffeting wind loading of structural members at an arbitrary attitude in the flow". In: *Journal of Wind Engineering and Industrial Aerodynamics* 56.2 (1995), pp. 267–290. ISSN: 0167-6105. DOI: https://doi.org/10.1016/0167-6105(94)00092-R.

[62] Bernardo Morais da Costa et al. "Bridge buffeting by skew winds: A quasi-steady case study". In: *Journal of Wind Engineering and Industrial Aerodynamics* 227 (2022), p. 105068.

[63] S. Benfratello, G. Falsone, and G. Muscolino. "Influence of the quadratic term in the alongwind stochastic response of SDOF structures". In: *Engineering Structures* 18.9 (1996), pp. 685–695. ISSN: 0141-0296. DOI: https://doi.org/10.1016/0141-0296(95)00222-7.

[64] Luping Yang, Kurtis R. Gurley, and David O. Prevatt. "Probabilistic modeling of wind pressure on low-rise buildings". In: *Journal of Wind Engineering and Industrial Aerodynamics* 114 (2013), pp. 18–26. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2012.12.014.

[65] J.D. Holmes. "Non-gaussian characteristics of wind pressure fluctuations". In: *Journal of Wind Engineering and Industrial Aerodynamics* 7.1 (1981), pp. 103–108. ISSN: 0167-6105. DOI: https://doi.org/10.1016/0167-6105(81)90070-2.

[66] Bo Li, Hiromasa Kawai, and Q.s Yang. "Non-stationary and non-Gaussian Characteristics of wind speeds". In: *Wind and Structures An International Journal* 24 (Jan. 2017), pp. 59–78. DOI: 10.12989/was.2017.24.1.059.

[67] Piergiovanni Marzocca, Jonathan M. Nichols, and Attilio Milanese. "An analytical formulation of bispectral densities for multiple degree-of-freedom systems". In: *Journal of Engineering Mathematics* 67 (2010), pp. 351–367. URL: https://api.semanticscholar.org/CorpusID:119963465.

[68] Li Cui et al. "Use of Bispectrum Analysis to Inspect the Non-Linear Dynamic Characteristics of Beam-Type Structures Containing a Breathing Crack". In: *Sensors* 21 (Feb. 2021), p. 1177. DOI: 10.3390/s21041177.

[69] Jonathan Nichols et al. "The Bispectrum and Bicoherence for Quadratically Nonlinear Systems Subject to Non-Gaussian Inputs". In: *Signal Processing, IEEE Transactions on* 57 (Nov. 2009), pp. 3879 –3890. DOI: 10.1109/TSP.2009.2024267.

[70] Massimiliano Gioffrè and Vittorio Gusella. "Numerical Analysis of Structural Systems Subjected to Non-Gaussian Random Fields". In: *Meccanica* 37 (Jan. 2002), pp. 115–128. DOI: 10.1023/A:1019666616309.

[71] Luca Caracoglia. "A stochastic model for examining along-wind loading uncertainty and intervention costs due to wind-induced damage on tall buildings". In: *Engineering structures* 78 (2014), pp. 121–132.

[72] K Aas-Jakobsen and E Strømmen. "Time domain buffeting response calculations of slender structures". In: *Journal of Wind Engineering and Industrial Aerodynamics* 89.5 (2001), pp. 341–364.

[73] Vincent Denoël and Maquoi René. "The concept of numerical admittance". In: *Archive of Applied Mechanics* 87 (2012), pp. 1337–1354.

[74] Vincent Denoël. "Polynomial approximation of aerodynamic coefficients based on the statistical description of the wind incidence". In: *Probabilistic Engineering Mechanics* 24.2 (2009), pp. 179–189. ISSN: 0266-8920. DOI: https://doi.org/10.1016/j.probengmech.2008.05.002.

[75] K Kimura and H Tanaka. "Bridge buffeting due to wind with yaw angles". In: *Journal of Wind Engineering and Industrial Aerodynamics* 42.1-3 (1992), pp. 1309–1320.

[76] Dorian Janjic and Heinz Pircher. "Consistent Numerical Model for Wind Buffeting Analysis of Long-Span Bridges". In: *IABSE Symposium Report*. Vol. 88. 6. International Association for Bridge and Structural Engineering. 2004, pp. 78–83.

[77] Robert H Scanlan. "Bridge buffeting by skew winds in erection stages". In: *Journal of engineering mechanics* 119.2 (1993), pp. 251–269.

[78] A. Kareem, M. A. Tognarelli, and K. R. Gurley. "Modeling and analysis of quadratic term in the wind effects on structures". In: *Journal of Wind Engineering and Industrial Aerodynamics* 74 (1998), pp. 1101–1110.

[79] Nadine Aubry, Régis Guyonnet, and Ricardo Lima. "Spatiotemporal Analysis of Complex Signals: Theory and Applications". In: *Journal of Statistical Physics* 64 (1991).

[80] A.G. Davenport and National Research Council Canada. Division of Building Research. *Wind Loads on Structures*. Technical paper - National Research Council of Canada, no. 088. National Research Council, 1960.

[81] Luigi Carassale, Giuseppe Piccardo, and Giovanni Solari. "Double Modal Transformation and Wind Engineering Applications". In: *Journal of Engineering Mechanics-asce - J ENG MECH-ASCE* 127 (May 2001). DOI: 10.1061/(ASCE)0733-9399(2001)127:5(432).

[82] Luigi Carassale, Giovanni Solari, and Federica Tubino. "Proper orthogonal decomposition in wind engineering - Part 2: Theoretical aspects and some applications". In: *Wind and Structures* 10 (2007). DOI: 10.12989/was.2007.10.2.177.

[83] Germund Dahlquist. "Convergence and stability in the numerical integration of ordinary differential equations". In: *Mathematica Scandinavica* (1956), pp. 33–53.

[84] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities". In: *International Journal for Numerical Methods in Engineering* 79 (Sept. 2009), pp. 1309 –1331. DOI: 10.1002/nme.2579.

[85] Jean-François Remacle et al. "A Frontal Delaunay Quad Mesh Generator Using the L∞ Norm". In: vol. 94. Jan. 2012, pp. 455–472. ISBN: 978-3-642-24733-0. DOI: 10.1007/978-3-642-24734-7_25.

[86] Paul George et al. *Mesh generation and mesh adaptivity: theory and techniques*. Jan. 2018.

[87] James D Foley et al. *Introduction to computer graphics*. Vol. 55. Addison-Wesley Reading, 1994.

[88] Peter Shirley, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. AK Peters/CRC Press, 2009.

[89] David Salomon. *Curves and surfaces for computer graphics*. Springer Science & Business Media, 2007.

[90] Idan Burstein. "Nvidia data center processing unit (dpu) architecture". In: *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE. 2021, pp. 1–20.

[91] Nick Sarkauskas et al. "Large-message nonblocking mpi_iallgather and mpi ibcast offload via bluefield-2 dpu". In: *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. IEEE. 2021, pp. 388–393.

[92] Georg Hager and Gerhard Wellein. *Introduction to high performance computing for scientists and engineers*. CRC Press, 2010.

[93] Alan Jay Smith. "Cache memories". In: *ACM Computing Surveys (CSUR)* 14.3 (1982), pp. 473–530.

[94] Charles Severance and Kevin Dowd. *High performance computing*. OpenStax CNX, 2010.

[95] Andrew Tanenbaum. *Modern operating system*. Pearson Education, Inc., 2009.

[96] Kévin Martins. *Bispectral-Analysis*. `https://github.com/ke-martins/bispectral-analysis`. 2022.

[97] Evan Davis. *random_data*. `https://github.com/emd/random_data`. 2018.

[98] Vincent Magnin et al. "Fortran... ok, and what's next?" In: *1024: Bulletin de la Société Informatique de France* (2024). URL: `https://hal.science/hal-04448657`.

[99] E. Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Pearson Education, 1994. ISBN: 9780321700698.

[100] Chris Lomont. "Introduction to intel advanced vector extensions". In: *Intel white paper* 23 (2011), pp. 1–21.

[101] Nigel Stephens et al. "The ARM scalable vector extension". In: *IEEE micro* 37.2 (2017), pp. 26–39.

[102] Utkarsh Ayachit. *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.

[103] Arthur Trew and Greg Wilson. *Past, present, parallel: a survey of available parallel computer systems*. Springer Science & Business Media, 2012.

[104] Robert Kuhn and David Padua. *Parallel Processing, 1980 to 2020*. Morgan & Claypool Publishers, 2020.

[105] Mark D Hill. "What is scalability?" In: *ACM SIGARCH Computer Architecture News* 18.4 (1990), pp. 18–21.

[106] OpenMP. *The OpenMP API specification for parallel programming*. 2023. URL: `https://www.openmp.org/` (visited on 03/19/2024).

[107] Rohit Chandra. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[108] Peter J Denning and Walter F Tichy. "Highly parallel computation". In: *Science* 250.4985 (1990), pp. 1217–1222.

[109] MPI Forum. *MPI Forum, standardization forum for the Message Passing Interface (MPI)*. URL: `https://www.mpi-forum.org/` (visited on 03/19/2024).

[110] Selim G Akl. *Parallel computation: models and methods*. Prentice-Hall, Inc., 1997.

[111] Min Si et al. "MT-MPI: Multithreaded MPI for many-core environments". In: *Proceedings of the 28th ACM international conference on Supercomputing*. 2014, pp. 125–134.

[112] Jon Peddie. *The History of the GPU - Steps to Invention*. Springer Nature, 2023.

[113] David B Kirk and W Hwu Wen-Mei. *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann, 2016.

[114] Manish Arora. "The architecture and evolution of cpu-gpu systems for general purpose computing". In: *By University of California, San Diego* 27 (2012).

[115] Bertil Schmidt et al. *Parallel programming: concepts and practice*. Morgan Kaufmann, 2017.

[116] Tor M Aamodt et al. *General-purpose graphics processor architectures*. Springer, 2018.

[117] Richard Vuduc and Jee Choi. "A brief history and introduction to GPGPU". In: *Modern Accelerator Technologies for Geographic Information Science* (2013), pp. 9–23.

[118] Michael McCool, James Reinders, and Arch Robison. *Structured parallel programming: patterns for efficient computation*. Elsevier, 2012.

[119] Jingyue Wu et al. "gpucc: an open-source GPGPU compiler". In: *Proceedings of the 2016 International Symposium on Code Generation and Optimization*. 2016, pp. 105–116.

[120] NVIDIA. *NVIDIA Corporation*. 2024. URL: `https://developer.nvidia.com/cuda-toolkit` (visited on 07/18/2024).

[121] Aaftab Munshi. "The opencl specification". In: *2009 IEEE Hot Chips 21 Symposium (HCS)*. IEEE. 2009, pp. 1–314.

[122] Aaftab Munshi et al. *OpenCL programming guide*. Pearson Education, 2011.

[123] Khronos Group. *The Khronos Group, Inc.* 2024. URL: `https://www.khronos.org/` (visited on 07/18/2024).

[124] Aksel Alpay and Vincent Heuveline. "SYCL beyond OpenCL: The architecture, current state and future direction of hipSYCL". In: *Proceedings of the International Workshop on OpenCL*. 2020, pp. 1–1.

[125] Ruyman Reyes et al. "Sycl 2020: More than meets the eye". In: *Proceedings of the International Workshop on OpenCL*. 2020, pp. 1–1.

[126] Rob Farber. *Parallel programming with OpenACC*. Newnes, 2016.

[127] Carlo Bertolli et al. "Integrating GPU support for OpenMP offloading directives into Clang". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. 2015, pp. 1–11.

[128] Brian W Kernighan and Dennis M Ritchie. *The C programming language*. Pearson Education Asia, 2002.

[129] *Information technology — Programming languages — C*. Standard. International Organization for Standardization, June 2018.

[130] *Programming languages - Fortran – Part 1: Base language*. Standard. International Organization for Standardization, Nov. 2023.

[131] Gautam Chakrabarti et al. "CUDA: Compiling and optimizing for a GPU platform". In: *Procedia Computer Science* 9 (2012), pp. 1910–1919.

[132] Wilson WL Fung et al. "Dynamic warp formation and scheduling for efficient GPU control flow". In: *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*. IEEE. 2007, pp. 407–420.

[133] Veynu Narasiman et al. "Improving GPU performance via large warps and two-level warp scheduling". In: *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. 2011, pp. 308–317.

[134] Khronos OpenCL Working Group et al. "The OpenCL specification version 1.1". In: *http://www. khronos. org/registry/cl/specs/opencl-1.1. pdf* (2011).

[135] Bureau d'Étude Greisch. *FINELG2020*. 2020.

[136] Michele Esposito Marzino and Vincent Denoël. *Analyse bispectrale d'une structure*. Tech. rep. Université de Liège, Apr. 2020.

[137] Michele Esposito Marzino and Vincent Denoël. *Analyse bispectrale d'une structure: résultats préliminaires*. Tech. rep. Université de Liège, Oct. 2020.

[138] Michele Esposito Marzino and Vincent Denoël. *Analyse Spectrale et Bispectrale d'une structure: résultats intérmediaires et finales*. Tech. rep. Université de Liège, Apr. 2021.

[139] Olivier Flamand and Vincent Denoël. "Influence of bridge deck shape on extreme buffeting forces". In: *Proceedings of 13th International Conference on Wind Engineering*. 2011.

[140] Michele Esposito Marzino et al. "Example of bispectral analysis of a transmission line pylon". In: *Proceedings of the XII International Conference on Structural Dynamics*. 2023.

[141] Gresich Design Office. *Le Viaduct De La Grande Ravine*. Tech. rep. 2009.

[142] Gresich Design Office. *Le Viaduct De La Grande Ravine À La Réunion*. Tech. rep. 2009.

[143] M Gioffrè, V Gusella, and M Grigoriu. "Simulation of non-Gaussian field applied to wind pressure fluctuations". In: *Probabilistic Engineering Mechanics* 15.4 (2000), pp. 339–345. ISSN: 0266-8920. DOI: https://doi.org/10.1016/S0266-8920(99)00035-1.

[144] Margaux Geuzaine. "Multiple Timescale Spectral Analysis of Wave-Loaded Floating Structures". PhD dissertation. Université de Liège, 2023.

[145] Margaux Geuzaine and Vincent Denoël. "Efficient estimation of the skewness of the response of a wave-excited oscillator". In: *Proceedings of the XI International Conference on Structural Dynamics*. 2020.

[146] Margaux Geuzaine et al. "Multiple timescale spectral analysis of floating structures subjected to hydrodynamic loads". In: *Journal of Engineering Mechanics* 149.3 (2023), p. 04023006.

[147] National Research Council Canada. Division of Building Research. *Wind Effects on Buildings and Structures: Proceedings [of The] International Research Seminar, Ottawa, Canada, 11-15 September 1967*. Wind Effects on Buildings and Structures: Proceedings [of The] International Research Seminar, Ottawa, Canada, 11-15 September 1967 v. 1. University of Toronto Press, 1968. ISBN: 9780802032133.

[148] Stephen A Rizzi, Adam Przekop, and Travis L Turner. "On the response of a nonlinear structure to high kurtosis non-Gaussian random loadings". In: *EURODYN2011-8th international conference on structural dynamics*. NF1676L-10853. 2011.

[149] Gal Berkooz, Philip Holmes, and John L. Lumley. "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows". In: *Annual Review of Fluid Mechanics* 25.1 (1993), pp. 539–575. DOI: 10.1146/annurev.fl.25.010193.002543.

[150] William K. George. "A 50-Year Retrospective and the Future". In: *Whither Turbulence and Big Data in the 21st Century?* Ed. by Andrew Pollard et al. Cham: Springer International Publishing, 2017, pp. 13–43. ISBN: 978-3-319-41217-7. DOI: 10.1007/978-3-319-41217-7_2.

[151] Y.C. LIANG et al. "Proper Orthogonal Decomposition and its applications — part i: theory". In: *Journal of Sound and Vibration* 252.3 (2002), pp. 527–544. ISSN: 0022-460X. DOI: https://doi.org/10.1006/jsvi.2001.4041.

[152] John L. Lumley. "The structure of inhomogeneous turbulent flows". In: *Atmospheric turbulence and radio wave propagation* (1967).

[153] Hans W. Liepmann. "Aspects of the Turbulence Problem". In: *Z. Angew. Math. Phys.* 3.5 (1952), 321–342. ISSN: 0044-2275. DOI: 10.1007/BF02008148.

[154] A. A. Townsend. *The Structure of Turbulent Shear Flow*. Cambridge University Press, 1956.

[155] Lawrence Sirovich. "Turbulence and the dynamics of coherent structures. I - Coherent structures." In: *Quarterly of Applied Mathematics* 45 (Oct. 1987). DOI: 10.1090/qam/910463.

[156] Luigi Carassale and Michela Marré Brunenghi. "Identification of meaningful coherent structures in the wind-induced pressure on a prismatic body". In: *Journal of Wind Engineering and Industrial Aerodynamics* 104-106 (2012). 13th International Conference on Wind Engineering, pp. 216–226. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2012.03.015.

[157] C. Picard and J. Delville. "Pressure velocity coupling in a subsonic round jet". In: *International Journal of Heat and Fluid Flow* 21.3 (2000), pp. 359–364. ISSN: 0142-727X. DOI: https://doi.org/10.1016/S0142-727X(00)00021-7.

[158] Ada Cammilleri et al. "POD-Spectral Decomposition for Fluid Flow Analysis and Model Reduction". In: *Theoretical and Computational Fluid Dynamics* 125 (Feb. 2013). DOI: 10.1007/s00162-013-0293-2.

[159] Bingchao Zhang et al. "Extended spectral proper orthogonal decomposition for analysis of correlated surrounding flow structures and wind load components of a building". In: *Journal of Wind Engineering and Industrial Aerodynamics* 240 (2023), p. 105512. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2023.105512.

[160] Oliver Schmidt and Tim Colonius. "Guide to Spectral Proper Orthogonal Decomposition". In: *AIAA Journal* 58 (Jan. 2020), pp. 1–11. DOI: 10.2514/1.J058809.

[161] Aaron Towne, Oliver Schmidt, and Tim Colonius. "Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis". In: *Journal of Fluid Mechanics* 847 (Aug. 2017). DOI: 10.1017/jfm.2018.283.

[162] Nick Schneider, Simon Köhler, and Jens von Wolfersdorf. "Experimental Detection of Organised Motion in Complex Flows with Modified Spectral Proper Orthogonal Decomposition". In: *Fluids* 8.6 (2023). ISSN: 2311-5521. DOI: 10.3390/fluids8060184.

[163] Peter J Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.

[164] Peter J Schmid et al. "Applications of the dynamic mode decomposition". In: *Theoretical and computational fluid dynamics* 25 (2011), pp. 249–259.

[165] Thomas Hawkins. "Cauchy and the spectral theory of matrices". In: *Historia Mathematica* 2.1 (1975), pp. 1–29. ISSN: 0315-0860. DOI: https://doi.org/10.1016/0315-0860(75)90032-4.

[166] Howard Anton and Chris Rorres. *Elementary Linear Algebra*. John Wiley New York, 2010.

[167] Nadine Aubry. "On the hidden beauty of the proper orthogonal decomposition". In: *Theoretical and Computational Fluid Dynamics* 2 (1991), pp. 339–352.

Appendix A

A general transformation process in a Finite Element model: from wind turbulence to wind forces

The first concerns the transformation of the wind turbulent components (i.e. turbulence velocities) into aerodynamic loads (i.e. forces), at the level of the set of the aerodynamic nodes `NNa`[1]. Then, such resulting loads, applied at the aerodynamic nodes, must be translated into aerodynamic loads directly applied to the structural nodes `NNs`. Therefore, Equation (4.16) could be rewritten, in a more general way, as:

$$\mathbf{f}(t) = \mathbf{XEa}^{(u)}\mathbf{u}(t) + \mathbf{XEa}^{(v)}\mathbf{v}(t) + \mathbf{XEa}^{(w)}\mathbf{w}(t) \tag{A.1}$$

in which $\mathbf{a}^{\mu_k}$ is a $\texttt{NNa} \times 1$ vector of wind model aerodynamic coefficients for each one of the three elementary wind turbulent component $\boldsymbol{\mu}_k$ ($\boldsymbol{\mu}_k \in \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$, $k = \{1, 2, 3\}$), $\mathbf{E}$ is a $\texttt{NNs} \times \texttt{NNa}$ matrix translating the forces at the aerodynamic nodes to equivalent forces at the structural nodes (see Figure A.1), and $\mathbf{X}$ is a $\texttt{NDOFs} \times \texttt{NNs}$ matrix accounting for the effect of each nodal force on the whole structural system. So that, the general matrix of wind model coefficients can be expressed as a composition of different transformations matrices:

$$\mathbf{A}^{(\mu_k)} = \mathbf{Ta}^{(\mu_k)}, \quad \boldsymbol{\mu}_k \in \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}, \; k = \{1, 2, 3\} \tag{A.2}$$

where

$$\mathbf{T} = \mathbf{XE} \tag{A.3}$$

[1]The concept of aerodynamic node embodies the simplest geometrical entity at which information concerning the wind flow can be asserted (i.e. measured).

is a `NDOFs` $\times$ `NNa` transformation matrix that groups the two transformation processes independent from the wind model, that translates wind turbulence speeds into wind forces.

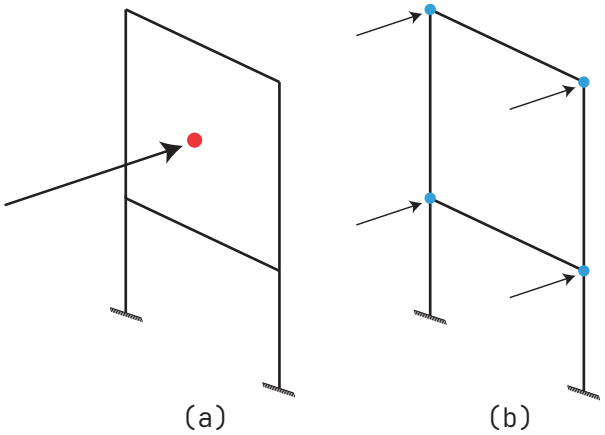


Figure A.1: Example of distribution of wind load. (a) Wind force at the aerodynamic node (red). (b) Equivalent force transformed at the enclosing structural nodes (blue).

The definition of $\mathbf{E}$ can be more or less intricate, based on the chosen interpolation method [73]. This is however not explicitly tackled in this Thesis, since its choice is simply a specialisation of more general concepts introduced in this section. To be noted however that, while this interpolation procedure might be relatively easy in a deterministic context (i.e. when finite realisation or generation of time series are available), the same cannot be implied in a stochastic context, when loading are only known in a probabilistic sense, hence the stochastic interpolation might indeed require different considerations with respect to the classical deterministic approach.
Nonetheless, most of the times, and for simplicity reasons, the set of aerodynamic nodes `NNa` is assumed to be a subset of the set of structural nodes, that is $\mathcal{N}_a \subset \mathcal{N}_s$, where $\mathcal{N}_s$ and $\mathcal{N}_a$ denote the set of structural and aerodynamic nodes respectively. Therefore, hereafter, the set of aerodynamic (and structural under such hypothesis) wind loaded nodes will be referred to as `NN` (i.e. dropping the `a` in the proper notation `NNa`, since now every aerodynamic node is indeed a loaded structural one). The set of all structural nodes will be still referred to as `NNs`[2]. This implies that ultimately, the matrix translating forces at the level of aerodynamic nodes into forces applied to structural nodes $\mathbf{E}$ can be split into two block matrices: a first `NN` $\times$ `NN` block which equals the identity matrix $\mathbf{I}$, and a second

[2]Since not all of them might possibly be under the wind action, e.g. a building with inner columns, protected from the wind action.

$(\texttt{NNs} - \texttt{NN}) \times \texttt{NN}$ block full of zeroes. In mathematical form:

$$\mathbf{E} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \tag{A.4}$$

The same can be done with the matrix $\mathbf{X}$ extending the effects of structural nodal forces to structural DOFs, so that Equation (A.3) can be rewritten as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{X}_a & \mathbf{X}_r \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} = \mathbf{X}_a \coloneqq \hat{\mathbf{X}} \tag{A.5}$$

since the remaining $\texttt{NNs} - \texttt{NN}$ structural nodes are not loaded (by the wind action), hence they will not contribute to the global forces applied to the set of structural DOFs.
Therefore, in such context, Equation (A.2) could be rewritten as

$$\mathbf{A}^{(\mu_k)} = \mathbf{T}\mathbf{a}^{(\mu_k)} \equiv \hat{\mathbf{X}}\mathbf{a}^{(\mu_k)} \tag{A.6}$$

where $\hat{\mathbf{X}}$ is the reduced $\texttt{NDOFs} \times \texttt{NN}$ matrix translating applied nodal forces to global structural DOFs actions (i.e. loadings). Finally, with these considerations Equation (A.1) takes the form

$$\mathbf{f}(t) = \hat{\mathbf{X}} \left(\mathbf{a}^{(u)}\mathbf{u}(t) + \mathbf{a}^{(v)}\mathbf{v}(t) + \mathbf{a}^{(w)}\mathbf{w}(t) \right) \tag{A.7}$$

Appendix B

Proper Orthogonal Decomposition: history and mathematical aspects

B.1 A historical review of POD

Starting from the first decades of the 20^{th} century, wind engineering as a branch of fluid dynamics started to arise and establish on a global scale.
The main object of interest was the study and characterisation of turbulence [149, 150]. In fact, even though the physical laws, the mathematical basis (in the Navier-Stokes equation of fluid mechanics) and experiments were already available, yet these tools were not enough for use in understanding this complex phenomenon. Some difficulties were in the non-linearity of such Navier-Stokes equations, and a very limited knowledge of their analytical solutions at high Reynolds numbers (i.e. in turbulent flows) [149, 150]. Therefore, this brought many authors at studying this phenomenon.
It is in such context that the Proper Orthogonal Decomposition (POD) arises. The concept of POD is a very broad mathematical concept. As a consequence, it goes under plenty of different names and acronyms. For example, POD can be identified under several different, but similar, methods: Principal Component Analysis (PCA), Karhunen-Loève Decomposition (KLD) and Singular Value Decomposition (SVD) [151], among all.
In wind related fields, many authors relate usage of POD techniques to the pioneering work introduced by Lumley[1] [152], with the goal of decomposing the

[1]Even though, such technique had been already proposed by many other authors in different disciplines [149]. For example, Solari in [43] pointed out that POD was initially born in the

random vector field representing a turbulent flow into a set of deterministic functions, each one of them capturing a portion of the total fluctuating kinetic energy in the flow. However, [150] points out that this misconception of POD being the initial work proposed by Lumley is what had actually prevented the field of turbulence studies evolve in the right direction, during second half of the 20^{th} century. Indeed, the two are somehow linked under certain assumptions.
To start simple, POD is a procedure that allows extracting a basis for a modal decomposition from an ensemble of random signals. The hope of such an innovative idea was in the fact that possibly a limited number of such deterministic functions (called the POD modes) would provide a very representative idea of the overall organisation of the turbulent flow. Moreover, POD would have helped finding the *coherent structures* (organised spatial features which repeatedly appear in time. Usually appearing in flows dominated by local shear) that made up the turbulent flow, but which were still hard to be found and observed. Such concept of coherent structures was firstly introduced by Liepmann in [153], and extensively used by Townsend in [154] ([149]), and later by Sirovich in [155]. The POD would have offered a rational method for extracting such flow features. The advantages in the development of such an approach were in the fact that, even though other methods existed already, each one of them had non negligible issues when applied to analytical studies of turbulence. Also, computational fluid dynamics (CFD) was at the time starting to emerge, as an alternative to the two conventional analytical approaches to the topic: statistical and deterministic analyses. The strength point of CFD was that it offered a direct resolution of the Navier-Stokes equations. However, the drawback was that such simulation provided little understanding of the solution it would have produced [149].
In that, POD would have provided solutions to all of those approaches [149]:

- It is statistically based
- Its analytical bases would help understand the method's strengths and limitations
- It allows to extract spatio-temporal structures considered essential based on some predetermined criteria, providing a rigorous mathematical framework for their complete description.

An important point of POD stands in the fact that it is a linear procedure, both in a positive and less positive view point. In fact, if in many applications POD (or PCA) has proved to perform well in extracting such coherent structures that resemble the observed physical phenomena, in some other applications it failed

XIX century under the name of Singular Value Decomposition (SVD), later transformed at the beginning of the 20^{th} century into PCA after Karhunen-Loève.

at providing meaningful results, mainly due to the orthogonality of the resulting modes [156]. As a consequence, a method using non-orthogonal modes has been formulated, named Independent Component Analysis (ICA). ICA assumes that measured data is derived from a generative model in which unknown sources (called Independent Components (IC)) are mixed, at each time instant, by an unknown matrix (referred to as mixing matrix). While in the POD it is assumed that modes (or PC in PCA) are orthogonal, ICA assumes that ICs are mutually statistically independent[2]. ICA coincides with POD (or PCA) for Gaussian random processes (or vectors) [43]. Nonetheless, linearity of POD does not imply any linear assumption on the problem involved.
An important variant of POD takes the name of spectral POD (SPOD) [157] It has been formulated after that it has been realised that classic POD applied to turbulent flows is good at capturing large coherent structures, but less at the micro structure (micro scale) level, since in such cases, these structures tend to curl in small spaces, hence their low energy content makes it difficult to be identified by POD [158, 159]. In this, SPOD is more advantageous since being it carried in the frequency domain, both the most energetic modes as the small scale turbulences (low kinetic energy, very localised in space) are identified at their characteristic frequencies. Also, SPOD modes represent coherent structures that evolve in space and time, while space-only POD modes generally do not [160, 161, 162]. Moreover, SPOD is in reality a special case of a more generalised spatio-temporal decomposition [79] under the assumption that the random process is statistically stationary [160], as it is the case of wind turbulence. So, if POD modes are the eigenvectors of the two-point spatial correlation functions at each time lag, SPOD modes are the eigenvectors of the cross-spectral density matrix (or tensor, CSD) at each frequency. Therefore, even though the "POD" part of SPOD is conceptually and algorithmically identical to the one of the classic POD (eigenvectors of a matrix), the "spectral" part is much less relatable. In fact, if one does not have already empirical methods for the estimation or evaluation of such cross-spectral density matrix of the random process, one has to meticulously use numerical methods for estimating it starting from time series realisations of the same random process (for example, Welch's method). This is what has been extensively investigated in [160], where authors discuss the discrete form of the SPOD (in the same spirit as the Karhunen-Loève decomposition is the discrete version of the continuous POD formulation), estimated from time series records. This is also what made SPOD gone a bit neglected in place of the classical space-only POD [160, 161, 162], or other Reduced Order Models (ROMs), as dynamic mode decomposition (DMD) [163, 164] and others, mainly due to the fact of (i) lack of a considerable amount time-resolved data, and (ii) difficulties in obtaining

[2]NOTE: in POD, POD coefficients (i.e. eigenvalues) are uncorrelated instead, carrying the total variance contribution of their relative POD mode (eigenfunctions, i.e. eigenvectors).

the cross-spectral densities for a large number of spatial points using hot wires, and large arrays of hot wires become intrusive, specially for small scale models [161]. Besides, SPOD would require very long time series and time-integrations that were almost prohibitive at the time of its formulation. Indeed, today things have significantly improved in that matter, and SPOD has come back to being used consistently in wind engineering applications.

It is in this light that the work carried in this Thesis is thought and developed. In the following chapters, POD techniques will be applied in a frequency domain approach, hence calling for SPOD. With the application of SPOD to the (PSDs of) wind turbulence field, spectra (and bispectra) of the wind load and structural responses will be found. Once these stochastic entities are computed, the relative statistical information (statistical moments) can be evaluated (by integrating these function in their domain of existence), finally leading to an estimate of the probability density function of the random process (see Figure 2.1).

B.2 A mathematical perspective to POD

In the most general fashion, POD refers to a very broad mathematical context. Indeed, POD is a well established and universal concept: find a set of ordered orthonormal basis functions (or vectors) in a subspace where a random process (vector) takes its values (i.e. its domain of existence), such that any sampled function (vector) in such space can be optimally expressed using this reduced subspace. The optimality is expressed in terms of least-square error on the variance, that is, the resulting variance is as close as possible to the exact one. To better understand, let's suppose to collect an ensemble of realisations of a given function $u(x)$ defined over a given domain X (i.e. $x \in X$). The problem stands in seeking a deterministic function ϕ that is most similar, on average, to the members of the ensemble. In mathematical terms, this reads

$$\psi = \max_{\psi} \frac{\mathbb{E}\left[|\left\langle u, \psi \right\rangle|^2\right]}{\left\langle \psi, \psi \right\rangle}, \tag{B.1}$$

where

$$\left\langle f, g \right\rangle = \int_X f(x) g^*(x) dx$$

denotes the function inner product, such that $\left\langle f, f \right\rangle^{1/2} = \sqrt{\int_X f(x) f^*(x) dx} = ||f||$ is the norm operator. To note that the average might be time, space, or ensemble averaging. The function $\phi(x)$ is found as that maximising the normalised inner product with the random function $u(x)$. A unitary normalised product would mean that the two functions are perfectly parallel in the function space.

Besides, in any of its forms, it relies on the computation of the eigenvectors and related eigenvalues of a given matrix, which usually goes under the name of *eigenvalue problem* [165, 149]. It consists, given a real matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, in finding a n-set of pairs $\lambda \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{R}^n$ such that:

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i \qquad \forall i = 1, \ldots, n \tag{B.2}$$

Clearly, the matrix $\mathbf{A}$ has to have some properties in order to ensure that Equation (B.2) admits a unique solution for all i in $\{1, \ldots, n\}$. Details on this aspect can be found in [166]. In this Thesis, it is implicitly assumed that such mathematical and geometrical conditions are met, for all matrices to which this methodology is be applied.

B.2.1 Karhunen-Loève decomposition

The Karhunen-Loève decomposition is the original and continuous form of the general POD concept. It was developed during the 1940s in the field of probability theory, as an optimal series expansion of continuous-time stochastic functions, or processes [151]. It states that a random function can be expanded as a series of deterministic functions with random coefficients. It is important to underline that it assumes the random function to be part of the L^2 family of scalar functions, i.e. it is square-integrable. Mathematically, it reads

$$u(x) = \sum_{i=1}^{\infty} a_k \phi_k(x) \tag{B.3}$$

where a_k are the random coefficients also called modal amplitudes, given by

$$a_k = \langle \phi_k(x) u(x) \rangle \,, \tag{B.4}$$

with $\phi_k(x)$ referring to the deterministic *empirical eigenfunctions* of the Fredholm integral equation

$$\int_X R(x, x') \phi_k(x') dx' = \lambda_k \phi_k(x), \tag{B.5}$$

where

$$R(x, x') = \mathbb{E}\left[u(x) u^*(x')\right] \tag{B.6}$$

is the two-point correlation tensor of the random function $u(x)$, In other words, from the Fredholm integral, it is clear that such eigenfunctions $\phi(x)$ are the eigenfunctions of its kernel, i.e. two-point correlation tensor R. The Fredholm equation formulation in Equation (B.5) is the continuous version of the eigenvalue problem formulated in Equation (B.2), where the matrix $\mathbf{A}$ is represented by the continuous two-point correlation function $R(x, x')$. To be noted that, the

averaging operation in Equation (B.6) is done over a given statistical variable in its probability space, in a general sense. The Karhunen-Loève decomposition can be easily extended to discrete processes of the form $x \in R^m$.
To give more context, it might be thought that such statistical variable can be time, space, or any other suitable variable related to the considered problem.

B.2.2 Space-only POD: vectorial form of the Karhunen-Loève decomposition

The extension of the KL decomposition to vectorial functions or vectorial fields, as for the case of wind turbulence, can be achieved by rewriting Equation (B.3) with bold notations, signifying vectorial quantities instead of scalar:

$$\mathbf{u}(\mathbf{x}) = \sum_{k=1}^{\infty} a_k \boldsymbol{\phi}_k(\mathbf{x}). \tag{B.7}$$

To distinguish from the original formulation, it is referred to as space-only POD. It is based on the concept of "snapshot" POD, where a snapshot can be seen exactly as taking a picture of a moving target. In this POD formulation, each snapshot is captured at a time scale distance such that two consecutive snapshot (i.e. images) can be considered time independent. When applied to fluid flows, POD provides some important properties, aside from the orthogonality of the basis vectors [160]:

- the ensemble of eigenvalues (λ_k) sum to (in the infinite case) the total variance of the process, i.e. the total kinetic energy of the wind flow. Mathematically:
$$\sum_k \lambda_k = E[\langle \mathbf{u}, \mathbf{u} \rangle] = E[\mathbf{u}^T \mathbf{u}]. \tag{B.8}$$

- Optimality. Any truncation of such a decomposition with n terms, captures more of the total variance than any other orthogonal expansion of the same order. This property is due to the ordering of eigenfunctions with decreasing eigenvalues $\lambda_k > \lambda_{k+1}$, such that any truncation will ensure keeping the first n eigenfunctions with the highest variance contributions.

- The random coefficients are uncorrelated, meaning that the expectation between any two different coefficients is zero
$$E[a_i a_j] = \lambda_i \delta_{ij} \tag{B.9}$$
where δ_{ij} represents the Kronecker delta. This comes from the orthogonality property of the relative eigenfunctions, or eigenvectors.

B.2.3 The Spatiotemporal Biorthogonal Decomposition

Introduced by Aubry in [167, 79], it extended the common practice of spatial-only POD. His approach is based on a double decomposition, in both space and time, whose resulting orthogonal modes are coupled: each space component called *topos* is associated with a time component, called *chronos*. The latter is the time evolution of the former, and the former is the spatial configuration of the latter. Mathematically, the biorthogonal decomposition formulates:

$$u(x,t) = \sum_{n=1}^{\infty} \lambda_n {\phi_k}^*(x)\psi_k(t). \tag{B.10}$$

One can clearly see the difference of Equation (B.10) with respect to Equation (B.3). In this approach, $\phi(x)$ and $\psi(t)$ are the eigenfunctions of two linear operators: for the former, the linear operator whose kernel is the two-point correlation function

$$\int_T R(x,x')\phi(x)dt = \lambda\phi(x) \tag{B.11}$$

where

$$R(x,x') = \int_T u(x,t)u^*(x',t)dt \tag{B.12}$$

while for the latter the linear operator whose kernel is the two-time (i.e. temporal) correlation function:

$$\int_X R(t,t')\psi(t)dx = \lambda\psi(t) \tag{B.13}$$

where

$$R(t,t') = \int_X u(x,t)u^*(x,t')dx. \tag{B.14}$$

The mathematical advantage of such bi-orthogonal decomposition is in the fact that it can be applied even to statistically non-stationary flows, thanks to Equation (B.14).
Nonetheless, it still keeps the two correlation concepts (time and spatial correlation) separate, not accounting for a possible unique correlation function accounting for both temporal and spatial shifts at the same time.

B.2.4 The revolutionising Lumley's work

Before starting, it should be noted that while this paragraph is probably the most important in the matter of POD introduction in turbulence fields, it is not exhaustive for those willing to dig deeper in the subject. It is however treated in this Thesis, not only for its fundamental importance, but also to ultimately

clarify the possible confusion that all the numerous approaches discussed up to now might have brought, recalling POD in one way or another.
In that, Lumley's work provides a generalisation of POD in the most broad sense. The work that he developed was based on a very simple and modest goal: find an objective way to identify the coherent structures, what Townsend in [154] calls "big eddy" [150].
The main question was: how could one chose a deterministic vectorial field that best represents and follows a given random field changing in both space and time? In mathematical terms, given a random vectorial field $u(\mathbf{x}, t)$ of space $\mathbf{x}$ and time t, how to find a deterministic function $\phi(\mathbf{x}, t)$ which optimally represents, in a mean square sense, the random field. This is done by maximising the square of their double inner product [150]:

$$\langle |u(\mathbf{x}, t)\phi(\mathbf{x}, t)| \rangle = \left\langle |\alpha|^2 \right\rangle \tag{B.15}$$

where

$$u(\mathbf{x}, t)\phi(\mathbf{x}, t) \equiv \int_X \cdots \int_{-\infty}^{+\infty} u(\mathbf{x}, t)\phi(\mathbf{x}, t) d\mathbf{x} dt \tag{B.16}$$

in which X denotes the entire spatial domain where the field is defined. Equation (B.16) defines the inner product in two senses: first, as a usual scalar product of two vectors, and secondly as a projection in Riemann space of two fields. This is what George in [150] refers to as the great Lumley's idea. From it, it would not be difficult, applying variational calculus, to obtain the final Lumley's formulation

$$\int_X \cdots \int_{-\infty}^{+\infty} R(\mathbf{x}, \mathbf{x}', t, t')\phi(\mathbf{x}', t') d\mathbf{x}' dt' = \lambda\phi(\mathbf{x}, t) \tag{B.17}$$

where its kernel $R(\mathbf{x}, \mathbf{x}', t, t') = \langle u(\mathbf{x}, t)u(\mathbf{x}', t') \rangle$ is the two-point two-time correlation, also called two-point two-time Reynolds stress tensor. Oftentimes, Equation (B.17) is referred as to the POD integral, in an erroneous way [150]. In fact, Equation (B.17) might reduce to it, but under certain circumstances. This happens only when the random process has the finite total energy property along all its spatial dimensions and time. This signifies that the integral of $R(\mathbf{x}, \mathbf{x}, t, t) = \langle u(\mathbf{x}, t)u(\mathbf{x}, t) \rangle$ is finite:

$$\int_X \cdots \int_{-\infty}^{+\infty} \langle u(\mathbf{x}, t)u(\mathbf{x}, t) \rangle \, d\mathbf{x} dt < \infty. \tag{B.18}$$

Important to note that this concept of energy finiteness does not imply the field to be finite, neither in space nor in time, just that its total energy is finite.
Indeed, if the field is limited in space or in time, the energy finiteness condition would be automatically satisfied, as soon as u is a physical bounded process.

So that is that only in such cases, results of Equation (B.17) can be referred to as POD solutions, and the sum of the eigenvalues equals the total (finite) energy in the field:

$$\int_X \cdots \int_{-\infty}^{+\infty} R(\mathbf{x}, \mathbf{x}, t, t) d\mathbf{x} dt = \sum_{n=1}^{\infty} \lambda_n. \tag{B.19}$$

Any random realisation of the field can then be represented as a linear combination of such eigenfunctions:

$$u(\mathbf{x}, t) = \sum_{n=1}^{\infty} a_n \phi_n(\mathbf{x}, t) \tag{B.20}$$

using some appropriate random coefficients given by projecting the field onto the eigenfunctions

$$a_n = \int_X \cdots \int_{-\infty}^{+\infty} \phi_n(\mathbf{x}, t) u(\mathbf{x}, t) d\mathbf{x} dt \tag{B.21}$$

while the eigenvalues are given by

$$\lambda_n = \langle a_n a_m \rangle \, \delta_{mn} \tag{B.22}$$

since the coefficients are uncorrelated.
However, there is one caveat. If each realisation of the instantaneous field can be reproduced from its POD decomposition, determining the coefficients requires the knowledge of the entire field at once, in space and at all times. This can be quite demanding in terms of physical storage required for such task, even with today's storage capacity, and therefore even more decades ago. In fact, determination of the eigenfunctions by Equation (B.17) requires only statistics, i.e. the determination of the two-point two-time correlation function R, which requires measurements at two space-time points. On the other hand, to reconstruct the instantaneous field from Equation (B.20), one needs the entire space-time field all at once, as shown Equation (B.21). This issue is also present in the spatiotemporal biorthogonal decomposition formulated by Aubry [79].

B.2.5 Spectral Proper Orthogonal Decomposition (SPOD)

Spectral proper orthogonal decomposition (SPOD) is one of the variants of POD that try to overcome the known issue of classical POD of losing time related information (i.e. spatial only POD modes), In fact, the major problem of Lumley's approach is that, for solutions of Equation (B.17) to be considered as effective POD solutions, the total energy of the process has to be finite (see Equation (B.18)). However, very few real flows have this property. Therefore, applying

this procedure to flows that are in reality of infinite energy, while ignoring this finite energy hypothesis, would result in "POD modes" which would be highly affected by such energy truncation, far from the real modes which would characterise the physical process. This truncation is usually caused by limiting the extents of the spatial domain, or by restricting simulation time.
There exists however a particular class of flow, which are homogeneous in space, and statistically stationary in time. Indeed, the homogeneity features makes them having an infinite energy, regardless of the time dimension. Therefore, as a consequence, in such cases, application of Lumley's integral would result in solutions that would be highly affected by the imposed boundaries, unless there exist some properties intrinsic to the kernel that make the integral to converge [150].
For that, the simple case of a one-dimensional field of infinite energy will be introduced first. Clearly, infinite energy is the result of fields being homogeneous and stationary. The stationarity property makes that each statistical quantity related to the process does not depend on the temporal origin at which it is computed. Similarly, homogeneity is the same concept, but in a spatial domain. Therefore, the two-time correlation function $R(t, t')$ will depend only on the time lag $\tau = t' - t$. Hence, Lumley's integral reduces to

$$\int_{-\infty}^{\infty} R(t,t')\phi(t')dt' = \int_{-\infty}^{\infty} R(\tau)\phi(t+\tau)d\tau = \lambda\phi(t), \tag{B.23}$$

$$\int_{-\infty}^{\infty} R(\tau)\left\{\frac{\phi(t+\tau)}{\phi(t)}\right\}d\tau = \lambda. \tag{B.24}$$

Equation (B.24) implies that the eigenfunctions $\phi(t)$ for a 1D homogeneous stationary random process are harmonic functions, with eigenvalues being their spectral energy content, given by the integration their Fourier transform in the frequency space. Following Equation (B.21), the coefficients end up being the Fourier transform of the field itself projected on the relative eigenfunction (i.e. harmonic function with radial frequency ω)

$$\hat{u}(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega t}u(t)dt. \tag{B.25}$$

Finally, any realisation of the random field can be reconstructed via an inverse Fourier transform

$$u(t) = \int_{-\infty}^{\infty} e^{i\omega t}\hat{u}(\omega)d\omega. \tag{B.26}$$

Now, it is quite straightforward to extend this procedure to the more general case of multiple spatial dimensions. The one dimensional two-time correlation function $R(\tau)$ generalises to the stationary version of the two-point two-time correlation

function

$$R(\mathbf{x}, \mathbf{x}', t, t') \rightarrow R(\mathbf{x}, \mathbf{x}', \tau)$$

, so that Equation (B.17) becomes

$$\int_X \cdots \int_{-\infty}^{+\infty} R(\mathbf{x}, \mathbf{x}', \tau)\phi(\mathbf{x}', t+\tau)d\mathbf{x}'d\tau = \lambda\phi(\mathbf{x}, t). \tag{B.27}$$

The cross-spectral density function (or matrix, or tensor) S can be defined as the Fourier transform of the correlation function

$$S(\mathbf{x}, \mathbf{x}', \omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} R(\mathbf{x}, \mathbf{x}', \tau)e^{-i\omega\tau}d\tau \tag{B.28}$$

while the correlation function is its inverse Fourier transform

$$R(\mathbf{x}, \mathbf{x}', \tau) = \int_{-\infty}^{+\infty} S(\mathbf{x}, \mathbf{x}', \omega)e^{i\omega\tau}d\omega \tag{B.29}$$

Applying the same mathematical concept as for the one-dimensional case, this implies that, for any radial frequency ω the function

$$\phi(\mathbf{x}, t) = \psi(\mathbf{x}, \omega)e^{i\omega t}$$

is a solution of the Lumley's integral (B.17), with eigenvalue equal to $\lambda(\omega)$, where functions $\psi(\mathbf{x}, \omega)$ and $\lambda(\omega)$ are the solutions of the spectral eigenvalue problem

$$\int_X S(\mathbf{x}, \mathbf{x}', \omega)\psi(\mathbf{x}', \omega)d\mathbf{x}' = \lambda(\omega)\psi(\mathbf{x}, \omega). \tag{B.30}$$

Equation (B.30) is the mathematical formulation of the spectral POD. The SPOD modes are the eigenfunctions or eigenvectors of the cross-spectral density (CSD) function or matrix of the random field.

Appendix C

Detailed mathematical formula of the bispectrum of a single element of the 3D matrix of bispectra of nodal forces

Following a detailed version [5] of the mathematical formula for the computation of an element of the 3D matrix of bispectra of nodal forces (see Section 4.7 and Equation (3.45)). It consists of a total of 27 terms, each one containing a product of two PSDs of wind turbulence (see Section 4.1).

$$
\begin{aligned}
B_{f_i f_j f_k}(\omega_1,\omega_2) = {} & 2a_i^{(5)}a_j^{(2)}a_k^{(2)}S_{u_I u_J}(\omega_1)S_{u_I u_K}(\omega_2)+ \\
& 2a_i^{(2)}a_j^{(5)}a_k^{(2)}S_{u_I u_J}(\omega_1+\omega_2)S_{u_J u_K}(\omega_2)+ \\
& 2a_i^{(2)}a_j^{(2)}a_k^{(5)}S_{u_I u_K}(\omega_1+\omega_2)S_{u_J u_K}(\omega_1)+ \\
& \\
& 2a_i^{(6)}a_j^{(3)}a_k^{(3)}S_{v_I v_J}(\omega_1)S_{v_I v_K}(\omega_2)+ \\
& 2a_i^{(3)}a_j^{(6)}a_k^{(3)}S_{v_I v_J}(\omega_1+\omega_2)S_{v_J v_K}(\omega_2)+ \\
& 2a_i^{(3)}a_j^{(3)}a_k^{(6)}S_{v_I v_K}(\omega_1+\omega_2)S_{v_J v_K}(\omega_1)+ \\
& \\
& 2a_i^{(7)}a_j^{(4)}a_k^{(4)}S_{w_I w_J}(\omega_1)S_{w_I w_K}(\omega_2)+ \\
& 2a_i^{(4)}a_j^{(7)}a_k^{(4)}S_{w_I w_J}(\omega_1+\omega_2)S_{w_J w_K}(\omega_2)+ \\
& 2a_i^{(4)}a_j^{(4)}a_k^{(7)}S_{w_I w_K}(\omega_1+\omega_2)S_{w_J w_K}(\omega_1)+ \\
& \\
& a_i^{(8)}a_j^{(3)}a_k^{(2)}S_{u_I u_K}(\omega_2)S_{v_I v_J}(\omega_1)+ \\
& a_i^{(2)}a_j^{(8)}a_k^{(3)}S_{u_I u_J}(\omega_1+\omega_2)S_{v_J v_K}(\omega_2)+ \\
& a_i^{(3)}a_j^{(2)}a_k^{(8)}S_{u_J u_K}(\omega_1)S_{v_I v_K}(\omega_1+\omega_2)+ \\
& a_i^{(8)}a_j^{(2)}a_k^{(3)}S_{u_I u_J}(\omega_1)S_{v_I v_K}(\omega_2)+ \\
& a_i^{(3)}a_j^{(8)}a_k^{(2)}S_{u_J u_K}(\omega_2)S_{v_I v_J}(\omega_1+\omega_2)+ \\
& a_i^{(2)}a_j^{(3)}a_k^{(8)}S_{u_I u_K}(\omega_1+\omega_2)S_{v_J v_K}(\omega_1)+ \\
& \\
& a_i^{(9)}a_j^{(4)}a_k^{(2)}S_{u_I u_K}(\omega_2)S_{w_I w_J}(\omega_1)+ \\
& a_i^{(2)}a_j^{(9)}a_k^{(4)}S_{u_I u_J}(\omega_1+\omega_2)S_{w_J w_K}(\omega_2)+ \\
& a_i^{(4)}a_j^{(2)}a_k^{(9)}S_{u_J u_K}(\omega_1)S_{w_I w_K}(\omega_1+\omega_2)+ \\
& a_i^{(9)}a_j^{(2)}a_k^{(4)}S_{u_I u_J}(\omega_1)S_{w_I w_K}(\omega_2)+ \\
& a_i^{(4)}a_j^{(9)}a_k^{(2)}S_{u_J u_K}(\omega_2)S_{w_I w_J}(\omega_1+\omega_2)+ \\
& a_i^{(2)}a_j^{(4)}a_k^{(9)}S_{u_I u_K}(\omega_1+\omega_2)S_{w_J w_K}(\omega_1)+ \\
& \\
& a_i^{(10)}a_j^{(4)}a_k^{(3)}S_{v_I v_K}(\omega_2)S_{w_I w_J}(\omega_1)+ \\
& a_i^{(3)}a_j^{(10)}a_k^{(4)}S_{v_I v_J}(\omega_1+\omega_2)S_{w_J w_K}(\omega_2)+ \\
& a_i^{(4)}a_j^{(3)}a_k^{(10)}S_{v_J v_K}(\omega_1)S_{w_I w_K}(\omega_1+\omega_2)+ \\
& a_i^{(10)}a_j^{(3)}a_k^{(4)}S_{v_I v_J}(\omega_1)S_{w_I w_K}(\omega_2)+ \\
& a_i^{(4)}a_j^{(10)}a_k^{(3)}S_{v_J v_K}(\omega_2)S_{w_I w_J}(\omega_1+\omega_2)+ \\
& a_i^{(3)}a_j^{(4)}a_k^{(10)}S_{v_I v_K}(\omega_1+\omega_2)S_{w_J w_K}(\omega_1)
\end{aligned}
\tag{C.1}
$$

Appendix D

Detailing of `BsaLib` public Application Programming Interface

This appendix details the public API (Application Programming Interface) of BsaLib, the library code developed in this Thesis.
Its structure is inspired to that of OpenCL[1], where each API call changes a private internal state, to be set properly before the main API call that runs the computational core is invoked. All BsaLib API calls can be grouped into these macro categories:

1. Initialisation and state setting;
2. Main core run;
3. Post-processing.

A schematic representation of such grouping can be visualised in Figure 5.12.
Nonetheless, contrarily to OpenCL, all API calls are handled synchronously. This is mostly due to the fact that in its interfacing layer, BsaLib tries to minimise any memory copy, hence only holding references (i.e. memory addresses) to any data coming from the user. The only data that might be copied are scalars, for which the copy is at most as expensive as taking a reference to the original data. This choice not only guarantees fast API calls, but it also aims at minimising the memory footprint, limiting any useless memory duplicates.
All API calls that are listed in this appendix can be found in the main interface module file, `BsaLib.F90`. Some API calls have been omitted, specially those

[1] `https://www.khronos.org/opencl/`

whose interface is clear enough to state their purpose. For the full list of BsaLib API call, visit the official repository documentation page[2].

D.1 General purpose API calls

D.1.1 `bsa_printBSAHeader`

```
module subroutine bsa_printBSAHeader()
end subroutine
```

Prints the `B.S.A.` ASCII logo to the console

```
 ---------------------------------------------
|      _____          ____                     |
|       /   \        /              /\         |
|      /____/        \___           /  \        |
|     /    \             \         /____\       |
|    /_____/  .    _____/  .    _/       \_  .  |
|---------------------------------------------|
```

D.1.2 `bsa_enableGPU`

```
module subroutine bsa_enableGPU()
end subroutine
```

Enables GPU offloading.

D.1.3 `bsa_doValidateModalData`

```
module subroutine bsa_doValidateModalData(bool)
   logical, intent(in) :: bool
end subroutine
```

Controls modal data validation:

`.true.`: enables modal data validation. It consists of eliminating all those vibration modes that are not 1-normalised. This explains why BsaLib differentiates between `NM` the total number of vibration modes (as per the user provision) and `NM_EFF`, which is in fact the total number of vibration modes kept after validation.

`.false.`: does nothing, and keeps modal data unchanged.

[2] `https://html-preview.github.io/?url=https://github.com/miEsMar/BsaLib/blob/main/doc/index.html`

☞ This API call was included in development stages, to separate vertical/horizontal modes from torsional ones, which are usually normalised differently (torsional modes are normalised based on unitary rotations, not displacements). This API call has to be considered **deprecated**, and will be eventually eliminated from future implementations.

D.1.4 bsa_generateBSAInputFiles

```
module subroutine bsa_generateBSAInputFiles(run)
   logical, value :: run
end subroutine
```

Enables generation of the built-in `BSA` executable compatible input files. It takes in input a logical variable, named `run`:

`.false.`: at run, generates `BSA` compatible files, and exits.

`.true.`: at run, generates files but continues normal execution.

D.1.5 bsa_forceBsaClsExecution

```
module subroutine bsa_forceBsaClsExecution(bool)
   logical, intent(in) :: bool
end subroutine
```

If `.true.` is passed, enables the forces execution of the CLASSIC approach.

D.1.6 bsa_Init

```
module subroutine bsa_Init()
end subroutine
```

Initialises BSALIB internal state.

☞ This API call should be preferably come before every other BSALIB related API call.

D.1.7 bsa_Run

```
module subroutine bsa_Run(m2mf_cls, &
    m2mr_cls, m2o2mr_cls, m3mf_msh, m3mr_msh, m3mf_cls, m3mr_cls)
  real(bsa_real_t), target, allocatable, dimension(:) :: m2mf_cls
  real(bsa_real_t), target, allocatable, dimension(:) :: m2mr_cls
```

```
  real(bsa_real_t), target, allocatable, dimension(:) :: m2o2mr_cls
  real(bsa_real_t), target, allocatable, dimension(:) :: m3mf_msh
  real(bsa_real_t), target, allocatable, dimension(:) :: m3mr_msh
  real(bsa_real_t), target, allocatable, dimension(:) :: m3mf_cls
  real(bsa_real_t), target, allocatable, dimension(:) :: m3mr_cls
end subroutine
```

D.1.8 bsa_Finalise

```
module subroutine bsa_Finalise()
end subroutine
```

Cleans BsaLib internal memory.

> ☞ This API call should be the last of all the BsaLib API calls.

D.1.9 bsa_isCleaned

```
logical pure module function bsa_isCleaned()
end function
```

Queries if BsaLib internal memory has been freed or not.

> ☞ Since this is query API call, it is safe to invoke it after `bsa_Finalise`.

D.2 API calls related to general BsaLib settings

D.2.1 bsa_setAnalysisType

```
module subroutine bsa_setAnalysisType(isuban)
   integer(bsa_int_t), value :: isuban
end subroutine
```

Sets type of analysis. Valid value for `suban`:

1: Classic analysis (DEFAULT).

2: Mesher analysis.

3: both.

D.2.2 bsa_setClassicMode

```
module subroutine bsa_setClassicMode(i_mode)
   integer(bsa_int_t), value :: i_mode
end subroutine
```

Sets computation mode for the Classic analysis type. Valid options for `i_mode` are:

1 `(BSA_CLASSIC_MODE_VECTOR):` the *vectorised* implementation used. This is indeed the preferred option in terms of speed. There is however a limitation of this approach: since it requires a considerable amount of allocated memory, and considered the limit of memory that is requirable to the runtime (before going to unoptimised mechanisms, such swap partition) if a given limit ($\approx$ 8 Gb) is exceeded, the BsaLib runtime automatically switches to a *scalar* implementation.

2 `(BSA_CLASSIC_MODE_SCALAR):` The internal *scalar* implementation is used. While this implementation is slower than its vectorised counterpart, it is certainly the one to be used to limit the memory footprint. Also, for very big cases, the BsaLib runtime might automatically switch to this implementation is too much memory allocation is required.

D.2.3 bsa_setScalingConv

```
module subroutine bsa_setScalingConv(iconv)
   integer(bsa_int_t), value :: iconv
end subroutine
```

Sets the Spectra scaling convention. Valid values for `iconv` are:

1 `BSA_PSD_CONVENTION_FREQ:` The *frequency* convention is used. In this case, the variance is given by the integration of the PSD over the positive range of frequencies:

$$m_2 = \sigma^2 = \int_0^{+\infty} S(f)df. \tag{D.1}$$

2 `BSA_PSD_CONVENTION_PULS:` The *circular frequency* convention is used. In this case, the variance is given by the integration of the PSd over the full range of circular frequencies:

$$m_2 = \sigma^2 = \int_{-\infty}^{+\infty} S(\omega)d\omega \tag{D.2}$$

☞ BsaLib by default uses the second convention over the circular frequencies.

D.2.4 `bsa_setSpectraComputation`

```
module subroutine bsa_setSpectraComputation(ipsd, ibisp)
   integer(bsa_int_t), value :: ipsd
   integer(bsa_int_t), value :: ibisp
end subroutine
```

If any of the two inputs `ipsd, ibisp` is 1, activates computation of the relative spectra (PSDs or bispectra).

D.2.5 `bsa_setSpectraExtension`

```
module subroutine bsa_setSpectraExtension(ionlydiag)
   integer(bsa_int_t), value :: ionlydiag
end subroutine
```

If `ionlydiag` is 1, only main-diagonal elements of both 2D and 3D tensors of PSDs and bispectra are computed. Else, if 0 is passed, all elements are computed. This is the default behaviour if this API call is not invoked.

☞ While computing only diagonal elements of the spectral tensors is indeed good in terms of computational performances (CPU time, since much less arithmetic operations), precision of final results, i.e. statistical moments, is highly affected, specially for higher (from third on) moments. Hence, `ONLYDIAG` kind of analyses are strongly discouraged.

D.2.6 `bsa_setupClassic`

```
module subroutine bsa_setupClassic(nfreqs, df)
   integer(bsa_int_t), value :: nfreqs
   real(bsa_real_t), value :: df
end subroutine
```

Provides main setting data for the Classic approach.

`nfreqs`: integer that specifies the number of discretisation points in the interval $[0, f_{max}]$, where f_{max} is determined as

$$f_{max} = (nfreqs - 1) * df.$$

Hence, if *circular frequencies* convention is used, this number will be internally actualised to the interval $[-f_{max}, f_{max}]$. The definition of f_{max} is done before the actualisation.

`df:` delta frequency Δf between each discretisation point.

D.2.7 `bsa_setupMesher`

```
module subroutine bsa_setupMesher(isvd, bkgrfmt, &
      bkgaext, genpaext, maxaext, ifcov, idumpmod)
   integer(bsa_int_t), value :: isvd
   integer(bsa_int_t), value :: bkgrfmt
   real(bsa_real_t),   value :: bkgaext
   real(bsa_real_t),   value :: genpaext
   real(bsa_real_t),   value :: maxaext
   integer(bsa_int_t), value :: ifcov
   integer(bsa_int_t), value :: idumpmod
end subroutine
```

Provides main setting data for the Mesher approach.

`isvd:` If 1, enables usage of POD. See Section 4.8.

`bkgrfmt:` specifies the base refinement (number of discretisation points, per side) of the background zone.

`bkgaext:` Defines the factor by which the background zone is extended (or reduced). The computation of the background zone base width is discussed in Section D.6.4. If a value > 1 is provided, this allows the user to extend this zone, avoiding cutting the zone's extensions where gradients might be still important. If a value < 1 is provided instead, the zone extension is reduced.

`genpaext:` Defines the factor by which the any peak zone is extended/reduced.

`maxaext:` Defines the factor by which the total covered area f_{max} is extended. In this case, this is done to avoid loosing information coming from the secondary peaks, which are usually placed at extensions up to $2 \cdot f_i$, where f_i is the $i-$th modal frequency. Values < 1 should be strongly avoided.

`ifcov:` deprecated.

`idumpmod:` If set to 1, includes modal data in the `dumpfile`.

D.3 API calls to control bispectral symmetries

D.3.1 `bsa_setSpatialSymmetry`

```
module subroutine bsa_setSpatialSymmetry(isym)
   integer(bsa_int_t), value :: isym
end subroutine
```

Controls the spatial symmetry of a bispectrum in the 2D frequency space. Valid values for `isym` are:

0: the full spatial information is computed. Reference case.

2: only half of the spatial information is computed.

4: only 1-fourth of the information is computed.

D.3.2 `bsa_setSpectraSymmetries`

```
module subroutine bsa_setSpectraSymmetries(ispctrsym)
   integer(bsa_int_t), value :: ispctrsym
end subroutine
```

Controls whether Spectra and Bispectra tensor symmetries are exploited or not. Valid values for `ispctrsym` are:

0: no symmetry used, all tensor elements are computed (see Figure D.1-a).

1: only symmetric elements are computed (see Figure D.1-b).

D.4 API calls related to wind data

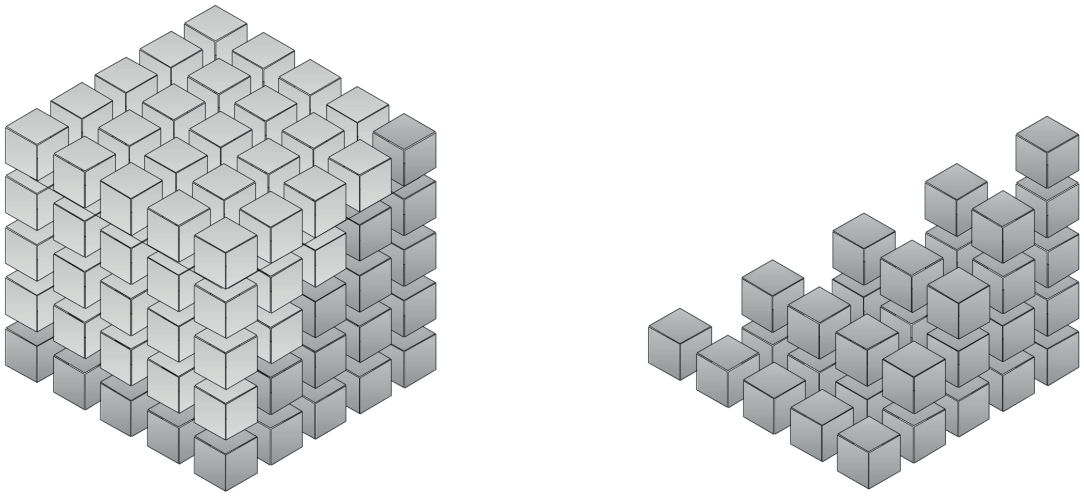
D.4.1 `bsa_setWindTurbComps`

```
module subroutine bsa_setWindTurbComps(tc, ntc)
   integer(bsa_int_t), intent(in) :: tc(:)
   integer(bsa_int_t), value, optional :: ntc
end subroutine
```

Specifies which turbulent components, among $\{u, v, w\}$, should be considered in the definition of wind loads (and for which turbulence PSDs will be computed).

`tc:` Array of components. Each array element value should be equal to one of $\{1 : u,\ 2 : v,\ 3 : w\}$.

`ntc:` total n. of turbulent components to consider.



(a) (b)

Figure D.1: (a) case where all the elements of the 3D tensor of bispectra are computed; (b) only the elements under the main diagonal are computed, the missing are inferred from symmetric elements.

D.4.2 `bsa_setWindVertProf`

```
module subroutine bsa_setWindVertProf(iwprof)
   integer(bsa_int_t), value :: iwprof
end subroutine
```

Sets the wind vertical profile. Valid options for `iwprof`:

1 `BSA_WIND_VERT_PROFILE_POWER` Mean wind speed is computed via a power law

$$\overline{U}(z) = \left(\frac{z}{z_{ref}}\right)^{\alpha} \tag{D.3}$$

where z indicates the altitude, z_{ref} the reference altitude, α the power profile coefficient.

NOTE: in most cases (see [48]), if $z < z_{ref}$, the value of $\overline{U}$ is taken equal to the value measured at the reference height.

2 `BSA_WIND_VERT_PROFILE_LOG`**:** A logarithmic law is used:

$$\overline{U}(z) = \frac{1}{k}\sqrt{\frac{\tau_0}{\rho}}\ln\frac{z}{z_0} \tag{D.4}$$

D.4.3 bsa_setPSDType

```
module subroutine bsa_setPSDType(ipsd)
   integer(bsa_int_t), value :: ipsd
end subroutine
```

Sets type of turbulence PSD. Valid options are:

`BSA_WIND_PSD_VONKARMAN`

`BSA_WIND_PSD_KAIMAL`

`BSA_WIND_PSD_DAVENPORT`

D.4.4 bsa_setAirDensity

```
module subroutine bsa_setAirDensity(aird)
   real(bsa_real_t), value :: aird
end subroutine
```

Sets the value of the air density ρ. By default, 1.225 $\frac{kg}{m^3}$.

D.4.5 bsa_setWZMeanWindVel

```
module subroutine bsa_setWZMeanWindVel(ubwz)
   real(bsa_real_t), target, intent(in) :: ubwz(:)
end subroutine
```

Sets the mean wind speeds $\overline{U}$, measured at the reference heights, for each wind zone.

D.4.6 bsa_setWZRefAlt

```
module subroutine bsa_setWZRefAlt(Zref)
   real(bsa_real_t), target, intent(in) :: Zref(:)
end subroutine
```

Sets the reference altitude z_{ref} for each wind zone.

D.4.7 bsa_setTurbWindScales

```
module subroutine bsa_setTurbWindScales(L)
   real(bsa_real_t), target, intent(in) :: L(3, 3, *)
end subroutine
```

Sets the 3×3 matrix of wind turbulence scales $\mathbf{L}$

$$\mathbf{L} = \begin{Bmatrix} L_{ux} & L_{vx} & L_{wx} \\ L_{uy} & L_{vy} & L_{wy} \\ L_{uz} & L_{vz} & L_{wz} \end{Bmatrix}$$

for each wind zone.

D.4.8 `bsa_setTurbWindSDT`

```
module subroutine bsa_setTurbWindSDT(sigma)
   real(bsa_real_t), target, intent(in) :: sigma(3, *)
end subroutine
```

Sets the wind turbulence standard deviation σ (i.e. turbulence intensity, $I = \frac{\sigma}{\overline{U}}$), for each wind zone.

D.4.9 `bsa_setWindCorrCoeffs`

```
module subroutine bsa_setWindCorrCoeffs(ccoeffs)
   real(bsa_real_t), target, intent(in) :: ccoeffs(3, 3, *)
end subroutine
```

Sets the 3×3 matrix of coefficients

$$\mathbf{C} = \begin{Bmatrix} C_{ux} & C_{vx} & C_{wx} \\ C_{uy} & C_{vy} & C_{wy} \\ C_{uz} & C_{vz} & C_{wz} \end{Bmatrix}$$

used in the decreasing-exponential formulation of wind spatial coherence [2].

D.4.10 `bsa_setWindCorrExpnts`

```
module subroutine bsa_setWindCorrExpnts(cexpn)
   real(bsa_real_t), target, intent(in) :: cexpn(3, 3, *)
end subroutine
```

Sets the 3×3 matrix of exponent coefficients

$$\mathbf{P} = \begin{Bmatrix} p_{ux} & p_{vx} & p_{wx} \\ p_{uy} & p_{vy} & p_{wy} \\ p_{uz} & p_{vz} & p_{wz} \end{Bmatrix}$$

used in the decreasing-exponential formulation of wind spatial coherence.

D.4.11 `bsa_setNodalVel`

```
module subroutine bsa_setNodalVel(Unod)
   real(bsa_real_t), target, intent(in) :: Unod(:)
end subroutine
```

Sets the mean wind speeds $\overline{U}$ for all structural nodes.

> ☞ As it has been extensively explained and discussed in Chapter 4, there is a distinction between structural and aerodynamic nodes. Structural nodes are the nodes used to model (e.g. FEM) the system. Aerodynamic nodes are the most basic entity at which information about wind flow (i.e. velocity) can be discretised. Theoretically, they are two separate concepts. However, most of the times, all aerodynamic nodes are structural nodes as well. Nonetheless, not all structural nodes are forcefully loaded by the buffeting action, so that usually `NN` refers to the whole set of structural nodes, while `NNL` to the subset of those effectively loaded. As per the current implementation, this API call requires the mean wind speed to be computed at all structural nodes (`NN`). In a future implementation, mean wind speeds at only the effectively loaded nodes (`NNL`) will be required, for optimality, and avoid any confusion in the internal implementation.

D.4.12 `bsa_setNodalWindZones`

```
module subroutine bsa_setNodalWindZones(NodWZ)
   integer(bsa_int_t), target, intent(in) :: NodWZ(:)
end subroutine
```

Sets the index of wind zone to which every structural node belongs to.

D.4.13 `bsa_setNodalWindAltitudes`

```
module subroutine bsa_setNodalWindAltitudes(WnodAlt)
   real(bsa_real_t), target, intent(in) :: WnodAlt(:)
end subroutine
```

Sets the altitude z of each structural node.

D.4.14 `bsa_setSpatialNodalCorr`

```
module subroutine bsa_setSpatialNodalCorr(nodCorr)
   real(bsa_real_t), target, intent(in) :: nodCorr(:, :)
end subroutine
```

Sets the matrix of spatial coherence between structural nodes. The dimensions are $(\texttt{NN}^2 + \texttt{NN}/2) \times 3$, where `NN` is the number of structural nodes, 3 the number of spatial turbulence components (i.e. *elementary* turbulence components) u, v, w.

> ☞ This API requires a specific format for the memory layout of the correlation coefficients. In fact, $\texttt{NN}^2 + \texttt{NN}/2$ is the number of elements in the lower-triangular part of a square matrix, diagonal elements included. Moreover, elements must be contiguous in memory following the Fortran convention, which orders multi-dimensional arrays starting from the leading dimension.

D.4.15 `bsa_setWindFCoeffs`

```
module subroutine bsa_setWindFCoeffs(wfc)
   real(bsa_real_t), target, intent(in) :: wfc(:, :, :)
end subroutine
```

Sets the multi-dimensional array of wind coefficients (see Equation (4.24)). For memory access optimality patterns, the dimensions should be `NNDOFs` × `NDEGW` × `NNL`.

D.5 API calls related to structural data

D.5.1 `bsa_setTotalNumOfDOFs`

```
module subroutine bsa_setTotalNumOfDOFs(ndofs)
   integer(bsa_int_t), value :: ndofs
end subroutine
```

Sets the total number of structural degrees-of-freedom `NDOFs`.

D.5.2 `bsa_setNumOfNodalDOFs`

```
module subroutine bsa_setNumOfNodalDOFs(nndofs)
   integer(bsa_int_t), value :: nndofs
end subroutine
```

Sets the number of total degrees-of-freedom per node `NNDOFs`.

D.5.3 `bsa_setTotalNOfNodes`

```
module subroutine bsa_setTotalNOfNodes(nn)
   integer(bsa_int_t), value :: nn
end subroutine
```

Sets the total number of structural nodes `NN`.

D.5.4 `bsa_setLoadedNodalDOFs`

```
module subroutine bsa_setLoadedNodalDOFs(libs_l, nlibs_l)
   integer(bsa_int_t), intent(in), target, allocatable :: libs_l(:)
   integer(bsa_int_t), value, optional :: nlibs_l
end subroutine
```

Sets the list of node loaded degrees-of-freedom.

> ☞ Usually, all nodal degrees-of-freedom `NNDOFs` should be considered loaded, once a node is loaded itself. However, this API call gives the possibility to set a custom list of values in the range $[1, \ldots, \texttt{NNDOFs}]$.

D.5.5 `bsa_setLoadedNodes`

```
module subroutine bsa_setLoadedNodes(nodes_l, nn_l)
   integer(bsa_int_t), intent(in), target, allocatable :: nodes_l(:)
   integer(bsa_int_t), value, optional :: nn_l
end subroutine
```

Sets the list `nodel_l` of loaded nodes, of length `nn_l`.

> ☞ This API call is optional. If not called, BsaLib automatically sets the list of loaded nodes to match the set of all structural nodes `NN`.

D.5.6 `bsa_setModalInfo`

```
module subroutine bsa_setModalInfo(ndofs, nm, Phi, natf)
   integer(bsa_int_t), value :: ndofs, nm
   real(bsa_real_t), intent(in), target :: Phi(ndofs, nm), natf(nm)
end subroutine
```

Sets modal info. Specifically, sets the `NDOFs` $\times$ `NM` modal matrix $\mathbf{\Phi}$ and array of modal natural frequencies $\mathbf{f}$.

> ☞ This API call (as others) requires to explicitly pass arrays dimensions, as a way to check correctness and coherence of the homogeneous data

passed to BsaLib.

D.5.7 bsa_setKeptModalShapes

```
module subroutine bsa_setKeptModalShapes(modes)
   integer(bsa_int_t), intent(in) :: modes(:)
end subroutine
```

Allows to set a custom list of modes to be kept in the analysis. This API call is complementary to the one discussed in Section D.1.3

D.5.8 bsa_setModalMatrices

```
module subroutine bsa_setModalMatrices(nm, Mgen, Kgen, Cgen)
   integer(bsa_int_t), value :: nm
   real(bsa_real_t), intent(in), target, dimension(nm) :: Mgen, Kgen
   real(bsa_real_t), intent(in), target :: Cgen(nm, nm)
end subroutine
```

Sets modal mass, stiffness, and damping matrices respectively.

☞ As per the current implementation, only symmetric modal mass and stiffness matrices are considered, meaning that in fact, the modal matrix $\mathbf{\Phi}$ given in is obtained by solving the following Eigenvalue problem:

$$\mathbf{K\Phi} = \mathbf{M\Phi\Lambda}$$

where $\mathbf{K}$ and $\mathbf{M}$ are the structural stiffness and mass matrices, $\mathbf{\Lambda}$ a diagonal matrix containing the coefficients of the characteristic polynomial equation. In such cases, the resulting modal mass and stiffness matrices

$$\mathbf{K}^\star = \mathbf{\Phi}^T\mathbf{K\Phi}; \qquad \mathbf{M}^\star = \mathbf{\Phi}^T\mathbf{M\Phi}$$

will be diagonalised by $\mathbf{\Phi}$. Therefore, only the diagonal elements are required by BsaLib. On the other hand, the whole NM $\times$ NM matrix of modal damping $\mathbf{C}^\star$ is required.

D.5.9 bsa_setTotDamping

```
module subroutine bsa_setTotDamping(xsi)
   real(bsa_real_t), target, intent(in) :: xsi(:)
end subroutine
```

Sets array of modal total damping ratios $\boldsymbol{\xi}$.

D.5.10 `bsa_getUsedModeShapes`

```
pure module function bsa_getUsedModeShapes() result(modes)
   integer(bsa_int_t), allocatable :: modes(:)
end function
```

Helper function that returns the list of modes effectively used within BsaLib.

D.6 API calls related to the `Mesher` algorithm

D.6.1 `bsa_enableOnlyPremesh`

```
module subroutine bsa_enableOnlyPremesh()
end subroutine
```

If called, exits after Pre-Meshing phase, skipping the Post-Meshing phase.

D.6.2 `bsa_doValidateZoneDeltas`

```
module subroutine bsa_doValidateZoneDeltas(bool)
   logical, intent(in) :: bool
end subroutine
```

If a `.true.` value is passed, enables validation of zone's frequency deltas (i.e. local zone refinement) to adhere to zone's policy. To control policies of zone types, see Section D.6.3.

☞ This API call has effect only in the Post-Meshing phase.

D.6.3 `bsa_setPolicyIDValidationValues`

```
module subroutine bsa_setPolicyIDValidationValues(id, &
      i_bfm, j_bfm, i_brm, j_brm)
   integer(int32), value :: id
   integer(int32), value :: i_bfm
   integer(int32), value :: j_bfm
   integer(int32), value :: i_brm
   integer(int32), value :: j_brm
end subroutine
```

`id:` built-in policy index for which setting custom values.

`i_bfm - j_bfm:` factors representing the number of segments in which dividing local $i-$ and $j-$ zone sides (segments), determined in the first Pre-Meshing phase, for the bispectra of modal forces $B_p(\omega_1, \omega_2)$.

`i_brm - j_brm:` factors representing the number of segments in which dividing local $i-$ and $j-$ zone sides (segments), determined in the first Pre-Meshing phase, for the bispectra of modal responses $B_q(\omega_1, \omega_2)$.

NOTE: this refinement process strictly follows the one on the (modal) loading points. After that is done, and refined segments are defined, then this refinement can occur.

D.6.4 `bsa_setMaxBkgPeakRestriction`

```
module subroutine bsa_setMaxBkgPeakRestriction(bool)
   logical, intent(in) :: bool
end subroutine
```

The width of the background (i.e. *quasi-static*) peak is equal to:

$$W_{\text{bkg}} = \frac{\overline{U}}{L} \quad [\text{Hz}] \tag{D.5}$$

where $\overline{U}$ is the mean wind speed, L the turbulence length scale. This information is crucial for determining the width of the main background zone, from which the definition of many other zones is based. In a 3D-spatial turbulence, there are in total 9 turbulence scales (lengths), organised in a 3×3 matrix of turbulence scales:

$$\mathbf{L} = \begin{Bmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \mathbf{l}_3 \end{Bmatrix} = \begin{Bmatrix} L_{ux} & L_{vx} & L_{wx} \\ L_{uy} & L_{vy} & L_{wy} \\ L_{uz} & L_{vz} & L_{wz} \end{Bmatrix} \tag{D.6}$$

By default (`bool = .false.`),

$$L = \text{maxval}(\mathbf{L}).$$

If `.true.` is passed instead, then the maximum value of the matrix is reduced to the maximum value along the first column:

$$L = \text{maxval}(\mathbf{l}_1),$$

that is turbulence scales along only the principal wind direction (X in the WRS) is considered.

D.6.5 `bsa_setPODTruncationThreshold`

```
module subroutine bsa_setPODTruncationThreshold(rval)
   real(bsa_real_t), value :: rval
end subroutine
```

Sets the total amount of energy to be kept from POD modes to `rval`. The value can be specified either in percentage ($[0, \ldots, 100]$), or absolute values ($[0, \ldots, 1]$).

D.6.6 `bsa_setPODNOfModesKept`

```
module subroutine bsa_setPODNOfModesKept(nmodes)
   integer(bsa_int_t), value :: nmodes
end subroutine
```

Sets desired number of POD modes to be kept. `nmodes` is a number between $[1, \ldots,$ `NNL`$]$, where `NNL` is the total number of loaded nodes.

> ☞ Since `bsa_setPODTruncationThreshold` has higher precedence, to make this API call effective, be sure not to invoke `bsa_setPODTruncationThreshold` as well.

D.7 API calls related to post-process

D.7.1 `bsa_enableVisualMode`

```
module subroutine bsa_enableVisualMode()
end subroutine
```

If this API procedure is called, enables *visual* mode. Visual mode allows the user to export (modal or nodal) bispectra to file, to be used in a post-processing (e.g. visualisation) stage.
When in visual mode, Pre-Meshing phase is skipped, and data is read from `dumpfile`.

> ☞ At the moment, this API call has effects only for the MESHER algorithm.

> ☞ This API call allows the exporting of one single bispectra only. In a future implementation, multiple bispectra must be allowed. See next

section D.7.2 for details on how the final index is computed.

D.7.2 `bsa_setVisualIndexes`

```
module subroutine bsa_setVisualIndexes(indexes, modal)
   integer(bsa_int_t), intent(in) :: indexes(3)
   logical, value :: modal
end subroutine
```

If `bsa_enableVisualMode` is called (see Section D.7.1), specifies base indexes from which computing the final index of the single bispectra to export.
It accepts 2 inputs:

`indexes:` array of 3 integers, denoting the base indexes from which computing the final index of the bispectra to export. This is computed differently whether is a bispectra of modal or nodal responses is desired.

If modal:

$$id = indexes(1) + indexes(2) * \texttt{M} + indexes(3) * \texttt{M}^2 \tag{D.7}$$

where each element of `indexes` is contained in the range $[1, \ldots, \texttt{M}]$, where `M` is the number of effective modes (`NM_EFF`) used within BsaLib.

If nodal:

$$id = (indexes(1) - 1) * \texttt{NNDOFs} + indexes(2) \tag{D.8}$$

where $indexes(1)$ refers to the *node* index, `NNDOFs` the number of degrees-of-freedom per single node, $indexes(2)$ the nodal DOF for which extracting the bispectral information.

`modal:` logical that specifies if the indexes refer to modal (`.true.`), or nodal (`.false.`).

☞ If only main-diagonal elements are computed, then the index is simply taken as the first index in the `indexes` array, that is:

$$id = indexes(1) \tag{D.9}$$

for both modal and nodal cases.

D.7.2.1 `bsa_setBRMExportDefaultMode`

```
module subroutine bsa_setBRMExportDefaultMode(imode)
   integer(bsa_int_t), value :: imode
end subroutine
```

Valid values for *imode*:

- `BSA_EXPORT_BRM_MODE_NONE`: exporting disabled
- `BSA_EXPORT_BRM_MODE_BASE`: using built-in exporting functionality.
- `BSA_EXPORT_BRM_MODE_USR` using custom user-specific exporting method.

  ☞ In such case, the actual procedure callback must be provided by calling `bsa_setBispExportCallback` (see Section D.7.2.2).

☞ Unlike other post-processing API calls, this procedure enables post-processing of structural data when `visual_mode` is OFF, i.e. when doing actual computation of statistical moments (see Section 5.3.3). Use with care, since this might considerably slow down even further execution time.

D.7.2.2 `bsa_setBispExportCallback`

```
module subroutine bsa_setBispExportCallback(fptr)
   procedure(exportInterf_vect_), pointer, intent(in) :: fptr
end subroutine
```

where `fptr` is a function pointer with the following interface

```
abstract interface
   subroutine exportInterf_vect_(f1, f2, brm, pdata)
      import :: bsa_real_t
      real(bsa_real_t), intent(in)  :: f1(:)
      real(bsa_real_t), intent(in)  :: f2(:)
      real(bsa_real_t), intent(in)  :: brm(:, :)
      class(*), pointer, intent(in) :: pdata
   end subroutine
end interface
```