



Nonlinear dynamic analysis tool development for contact interactions

Nicolas Leclère

► To cite this version:

Nicolas Leclère. Nonlinear dynamic analysis tool development for contact interactions. Mechanical engineering [physics.class-ph]. 2018. dumas-03258225

HAL Id: dumas-03258225

<https://dumas.ccsd.cnrs.fr/dumas-03258225>

Submitted on 11 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LIÈGE - FACULTÉ DES SCIENCES APPLIQUÉES

MASTER THESIS

Nonlinear Dynamic Analysis Tool Development for Contact Interactions

Author:

Nicolas LECLÈRE

Academic Supervisors:

Alain BATAILLY

Gaëtan KERSCHEN



GRADUATION STUDIES CONDUCTED FOR OBTAINING THE MASTER'S
DEGREE IN AEROSPACE ENGINEERING BY NICOLAS LECLÈRE

Academic Year : 2017-2018

Abstract

Nonlinearity is a common issue when examining cause-effect relationships. Such instances require complex modelling and hypothesis to offer explanations to nonlinear events. Nonlinear properties are present in many fields in Science and Engineering (optics, mechanics, ...) Among them in aerospace engineering, the description of the contact interactions between aircraft engine blades and the associated casing is of prime importance. In this context, the study deals with the development of an original versatile open source tool based on the combination of the two computational languages (Python and C++) able to model and simulate interactions between multiple structures. On one hand, by using *code_aster*, the mechanical analysis allows the application of boundary conditions, creation of materials and computation of eigen-modes. On the other hand, on top of the open source aspect, the tool has been developed to be as flexible as possible to allow the addition of further modules more adapted to specific analysis. From a computational point of view, the mechanical contact has been described by using two recognized methods, namely the Lagrange Multipliers and the Penalty methods. The time integration process has been conducted by using the well known algorithm based on the Newmark schemes.

To validate the code, the first test case considered corresponds to the interactions between a falling cube between two rings. All the pertinent computational and physical parameters were systematically compared and optimized. Depending on the initial conditions, different mechanical behaviours were observed.

The major contribution of this work was the analysis of the contact interaction in a more complex case of a compressor blade entering in contact with its casing during rotation. The algorithm allows to compare how the dynamics of the engine is impacted. The influence of the reduction method on the obtained results clearly proved its advantages on large structures and complex finite element models. Interesting results have been observed when the response at the leading edge and at the trailing edge were compared (both in time and frequency domain). For instance, by increasing the rotation velocity, the interaction between the response and the harmonics amplified the frequency response up to a certain point where the harmonics and the response are fused together. After what the response suddenly dropped.

This nonlinear behaviour retrieved thanks to the developed tool can be considered as a last validation proving its robustness.

Acknowledgements

First I would like to express my gratitude to Dr. Alain Batailly for receiving me in his laboratory at Polytechnique Montréal during six months and thanks to whom I worked on a interesting and motivating project. He provided me great support and shared his knowledge on many interesting topics that helped me to achieve my work.

I express my gratitude to Dr. Gaëtan Kerschen for his supervision from ULiège. Without him i would not have the opportunity to integrate this project to my master and represent the university abroad. I would also like to thank the university for their financial support which made this internship possible.

I am particularly thankful to Dr. Florence Nyssen for her constant support and supervision during this work. I am grateful to her and her companion who made me discover the country and helped me to settle when I first arrived.

I address my thanks to the jury members of this report for the time and effort they are putting on the evaluation of my work.

Finally, I thank my parents for giving me the opportunity to do my internship abroad. I also thank my family and friends who supported me during this work and my studies.

Table of contents

List of figures	ix
List of tables	xv
Nomenclature	xv
1 Introduction	1
1.1 Motivations & Objectives	1
1.2 Structure of the thesis	3
2 Theoretical Aspects	5
2.1 Introduction	5
2.2 Numerical integration - Newmark schemes	6
2.3 Reduction techniques	8
2.3.1 Principle	8
2.3.2 Guyan-Iron method [14]	9
2.3.3 Craig-Bampton method [9]	10
2.4 Surface interpolation procedures	10
2.4.1 Cox-de Boor formula	11
2.4.2 Bicubic b-spline surfaces	11
2.5 Contact treatment	18
2.5.1 Distance Computation using Newton-Raphson method	18
2.5.2 Contact forces	19
2.6 Conclusions	25
3 Notice and code presentation	27
3.1 Code_Aster & Finite Element Model	28
3.2 Python code: reduction and communication	29
3.3 C++ code: core of the numerical tool	33

4	Validation of the numerical tool	37
4.1	Presentation of the test case	38
4.1.1	Geometry and mechanical properties	38
4.1.2	Initial Velocity	39
4.1.3	Symmetry of the results	40
4.2	Convergence Analysis	41
4.2.1	Mesh of the rings	41
4.2.2	Modal Reduction Parameter of the cube	45
4.2.3	Time step	51
4.2.4	Newton-Raphson coefficient	53
4.3	Analysis of the results	54
4.3.1	Flexible Rings	54
4.3.2	Damping	57
4.3.3	Friction coefficient	59
4.4	Comparison between Penalty and Lagrange Multipliers methods	63
5	Interaction of contact between a compressor blade and its casing	69
5.1	Description of the simulation case	70
5.2	Analysis of the results	73
5.2.1	Rigid casing	73
5.2.2	Flexible casing	84
6	Conclusions	93
6.1	General Conclusions	93
6.2	Perspectives	95
	References	97

List of figures

1.1	Example of non-linear effects that can appear in an aircraft engine [24].	2
2.1	Newmark method stability regarding the parameters β and γ [12].	7
2.2	Transformation into a rectangular domain extracted from [3].	12
2.3	Flowchart of the Newton-Raphson for the distance computation.	20
2.4	Graphical representation of the Newton-Raphson iterations.	21
2.5	Flowchart of the contact treatment.	22
2.6	Graphical representation of the penetration of a node and its correction. . . .	23
2.7	Graphical comparison of Lagrange Multiplier method and Penalty method [7].	25
3.1	Logo code_aster	28
3.2	Typical example representing a cube interacting with two rings as a finite element model (a) and the equivalent representation in the tool (b).	32
3.3	Structure of the code.	35
4.1	Global configuration of the test case.	38
4.2	Finite element model of the cube falling between two rings.	39
4.3	Comparison of different falling velocities for the cube between two rigid rings.	40
4.4	Node of interest on the cube for the validation of the symmetry represented by the red dots.	40
4.5	Comparison of the displacement of the two nodes of the bottom face of the cube along X (a), Y (b) and Z (c). (—) Right Node, (- -) Left Node.	41
4.6	Comparison of the different meshes for the rings.	42
4.7	Node of interest on the cube indicated by a red dot.	43
4.8	Distance between the cube and the rings (a) and displacement of the cube along X (b), Y (c) and Z (d) for different meshes. (—, —) Very Coarse Mesh, (—, —) Coarse Mesh, (—, —) Fine Mesh, (—, —) Very Fine Mesh.	44
4.9	Reduced model of the cube.	45
4.10	Nodes of interest on the cube represented by the 4 red dots.	46

4.11	Distance between the cube and the infinitely rigid rings for different reduction parameters. (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.	46
4.12	Difference of distance between the reduced cube and the full one for different reduction parameters. (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes.	47
4.13	Displacement along X of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.	48
4.14	Displacement along Y of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.	49
4.15	Displacement along Z of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.	50
4.16	Comparison of the average computation time of one iteration with respect to the reduction parameter without contact (a) and during the contact (b).	51
4.17	Distance between the cube and the infinitely rigid rings for different time steps. (—) $h = 5 \cdot 10^{-6}$ s, (—) $h = 1 \cdot 10^{-6}$ s, (—) $h = 5 \cdot 10^{-7}$ s, (—) $h = 1 \cdot 10^{-7}$ s, (—) $h = 5 \cdot 10^{-8}$ s.	52
4.18	Zoom on the distance between the cube and the infinitely rigid rings for different time step. (—) $h = 5 \cdot 10^{-6}$ s, (—) $h = 1 \cdot 10^{-6}$ s, (—) $h = 5 \cdot 10^{-7}$ s, (—) $h = 1 \cdot 10^{-7}$ s, (—) $h = 5 \cdot 10^{-8}$ s.	53
4.19	Distance between the cube and the flexible rings (a). Displacement of the node of the cube along X (b), Y (c) and Z (d). (—) Rigid, (—) Flexible, (—) Rigid, (—) Flexible.	55
4.20	Comparison of the contact efforts, λ , for rigid rings (—) and flexible rings (—).	56
4.21	Displacement of one node of the flexible rings along X (a), Y (b) and Z (c) for different reduction parameters. (—) 0 constrained modes (—) 100 constrained modes, (—) 200 constrained modes.	57

4.22	Displacement of the node of the cube along X (a) and Z (b). (—) Damped, (—) No Damping.	59
4.23	Displacement of the node of the cube along X (a) and Z (b) Displacement of the node of the rings along X (c), Y (d), Z (e). (—, —) Damped, (—, —) No Damp.	60
4.24	Distance between the cube and the rings for different friction coefficient (a). Displacement of the node of the cube along X (b), Y (c) and Z (d) for different friction coefficient. (—, —) $\mu = 0$, (—, —) $\mu = 0.1$, (—, —) $\mu = 0.3$, (—, —) $\mu = 0.8$	61
4.25	Distance between the cube and the rings for different friction coefficient (a). Displacement of the node of the cube along X (b), Y (c) and displacement of the node of the ring along X for different friction coefficient (d). (—, —, —) $\mu = 0$, (—, —, —) $\mu = 0.1$, (—, —, —) $\mu = 0.3$, (—, —, —) $\mu = 0.8$	62
4.26	Distance between the cube and the rigid rings for the two contact methods (a). Contact forces for the two contact methods with rigid rings (b). Displacement of the node of the cube along X (c), Y (d) for the two contact methods with rigid rings. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Penalty $k_p = 10^9$, (—, —, —) Penalty $k_p = 10^{10}$, (—, —, —) Lagrange Multipliers.	63
4.27	Distance between the cube and the flexible rings for the two contact methods (a). Contact forces for the two contact methods with flexible rings (b). Displacement of the node of the cube along X (c), Y (d) for the two contact methods with flexible rings. Displacement of the node of the ring along X (e), Y (f) for the two contact methods with flexible rings. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Penalty $k_p = 10^9$, (—, —, —) Penalty $k_p = 10^{10}$, (—, —, —) Lagrange Multipliers.	65
5.1	Basic components of a gas turbine engine [21].	69
5.2	Finite element model of the compressor blade casing with the clamped nodes represented by yellow dots.	71
5.3	Finite element model of the turbine blade with the retained nodes represented by red dots and clamped nodes represented by yellow dots.	71
5.4	Convergence of the eigen-mode of the turbine blade with respect to the amount of constrained mode considered in the reduction. (—) 0 constrained-modes, (—) 5 constrained-modes, (—) 10 constrained-modes, (—) Full structure.	72

5.5	Representation of the retained nodes of the compressor blade inside the casing in the developed tool.	73
5.6	Distance between the leading edge of the compressor blade and the rigid casing for different contact algorithms (a). Contact forces at the leading edge of the compressor blade for different contact algorithms (b). Displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	75
5.7	Zoom on the distance between the leading edge of the compressor blade and the rigid casing for different contact algorithms (a). Zoom on the contact forces at the leading edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	76
5.8	Distance between the trailing edge of the compressor blade and the rigid casing for different contact algorithms (a). Contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	77
5.9	Zoom on the distance between the trailing edge of the compressor blade and the rigid casing for different contact algorithms (a). Zoom on the contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	78
5.10	Representation of the impacts position on the casing for the leading edge (a) and the trailing edge (b).	79
5.11	Fast Fourier Transform of the displacement along X of the leading edge (a) and trailing edge (b) for a rigid casing. (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.	80
5.12	Comparison of the FFT of the displacement along X of the trailing edge for different rotation speeds using the Lagrange Multipliers method.	81
5.13	View from the top of Figure 5.12.	82

5.14	Comparison of the FFT of the displacement along X of the trailing edge for different rotation speeds using the Penalty method with $k_p = 10^8$	82
5.15	View from the top of Figure 5.14.	83
5.16	Distance between the leading edge of the compressor blade and the flexible casing for different contact algorithms (a). Contact forces at the leading edge of the compressor blade for different contact algorithms (b). Displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	84
5.17	Zoom on the distance between the leading edge of the compressor blade and the flexible casing for different contact algorithms (a). Zoom on the contact forces at the leading edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	85
5.18	Distance between the trailing edge of the compressor blade and the flexible casing for different contact algorithms (a). Contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	86
5.19	Zoom on the distance between the trailing edge of the compressor blade and the flexible casing for different contact algorithms (a). Zoom on the contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.	87
5.20	Finite element model of the compressor blade casing with the studied nodes represented by red dots.	88
5.21	Displacement of the node of the flexible casing along X (a), Y (b) and Z (c) for different contact algorithms. (—) Penalty $k_p = 10^6$, (—) Penalty $k_p = 10^7$, (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.	89

5.22	Zoom on the displacement of the node of the flexible casing along X (a), Y (b) and Z (c) for different contact algorithms. (—) Penalty $k_p = 10^6$, (—) Penalty $k_p = 10^7$, (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.	90
5.23	Fast Fourier's Transform of the displacement along X of the leading-edge (a) and trailing-edge (b) for a flexible casing. (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.	90

List of tables

2.1	Newmark methods family.[15]	7
4.1	Mechanical properties of the cube and the two rings.	39
4.2	Difference of distances (in mm) between the different meshes and the finer mesh.	44
4.3	Computation time for the first iteration of the tool with different N-R coefficient.	54
5.1	Relative errors on each mode between the reduction for different reduction parameters and the full structure.	72

Chapter 1

Introduction

1.1 Motivations & Objectives

In the recent years there has been an increase in the research about non-linear effects in mechanics. While most of the non-linear component of studies have been neglected in the past, the increasing complexity of the structures design imposed nowadays their consideration. The increase of the computation resources have also greatly improve the construction of representative numerical model closer to actual design. The construction of complex model is not the only improvement that has been made possible thanks to the increase of the computation power. The efficiency of the algorithms such as the time integration or the various iterative procedures have gained from the resources improvement. The possibilities regarding the type of simulation have also grown with the resources development.

One can differentiate non-linear effects regarding their origins. The first category directly concerns the material itself. For instance hyper-elastic material, like the rubber, or shape memory alloy have a strong non-linear behaviour that must be taken into account. The friction is also a source of non-linearity. The second category is related to the excitation that the structure undergoes. Upon an external excitation it can induce a non-linear response. The third can be associated with the damping, and more specifically with the interfaces between the different components, for instance bolted joints induce non-linearities. The last category, which is the one of interest in this study, is related to geometric non-linearities. However, they can be separated into two distinct sub-categories. Either the non-linearities are due to geometry itself or they are due to the boundary conditions. For instance, due to their very thin structures, turbine blade, are showing non-linear behaviours once they are in rotation. This is considered as a geometric non-linearity but if a contact is existing, even temporary, between the turbine blade and the casing, one can observe non-linear behaviour due to the boundary

conditions. This work is focusing on this last type of interaction. Figure 1.1 represents an aircraft engine where some of the non-linear effects presented here are pointed out.

Several software for the analysis and the contact treatment are already existing, they often are the property of large companies or are eventually tied to licensed third party software. The objective of this project is to develop an open source laboratory numerical tool to model and simulate contact interactions between multiple structures.

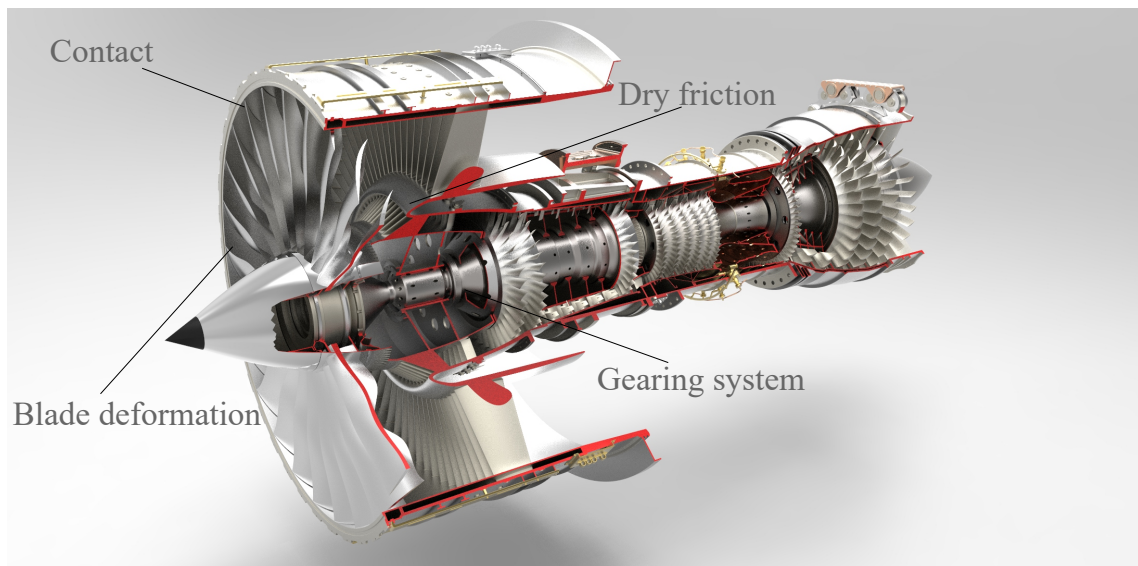


Fig. 1.1 Example of non-linear effects that can appear in an aircraft engine [24].

Additionally to the open source aspect, the software had to be easy to upgrade with future developments. This strongly guided the construction of the code from its very foundation, influencing the choice of computational language as well as the general architecture of the code.

The contact simulation has been conducted using two well known methods in the field of contact treatment. The Lagrange Multipliers and the Penalty method. The results associated to the two methods are compared in this report. The time integration has also been conducted using a well known method associated to the Newmark's schemes.

1.2 Structure of the thesis

Chapter 1 concerns the global theory that have been used in the development of the software used in the scope of this project. The time integration method as well as the numerical representation of the studied structures are explained. The algorithm considered for the contact treatment is also introduced.

Chapter 2 consists in a general presentation of the numerical tool and its usage. The different parts constituting the whole code including the interactions between those parts are introduced.

Chapter 3 is about the validation of different aspects of the numerical tool. The geometry of the test case and the initial conditions are first exposed. The dependencies of the results with respect to the numerical parameters such as the time step is then introduced. The third part of this chapter concerns the impact of the physical parameters, like the friction, on the results. Finally, the responses to the different contact algorithm are compared.

Chapter 4 tackles the contact interaction in a complex case of a turbine blade entering in contact with its casing during the rotation. The contact algorithm are both used to compare how the dynamic of the turbine blade is impacted.

Finally conclusions about the contact interactions are drawn and future development ideas are proposed.

Chapter 2

Theoretical Aspects

2.1 Introduction

In this second chapter, a general presentation of the theoretical aspects necessary for this project are presented.

The first section provides a detailed presentation of the time integration methods of the Newmark's family. Time integrations method are present in a wide range of applications including the work that is presented in this report. The stability and the numerical parameters of the methods are discussed.

The second section concerns the reduction methods. The principle of those methods and the equations governing them are presented. The Craig-Bampton method is discussed in particular.

The interface between the different structures is modelled thanks to a b-spline surface. It is in this context that the third section gives a practical presentation of the construction of b-spline surfaces. The algorithms used for the construction of a b-spline surface interpolating a set of points are described.

The fourth and final section presents the contact treatment. The construction of the surface conducts the complexity of the distance determination to increase. In this regard, a Newton-Raphson algorithm is used. The associated theoretical concepts are first introduced. In a second time, the Lagrange Multipliers method and the Penalty method are presented and briefly compared.

2.2 Numerical integration - Newmark schemes

The numerical modelling and simulation of real structures and physical phenomena requires to transform the continuous representation of time into a discrete parameter. There are more than one method to achieve this transformation. The one used in the scope of this project is a direct time integration based on finite time differences. It means that the time step is a critical parameter that strongly affects the stability of the analysis. A famous method for the time integration is known as the Newmark method [15, 17]. This method is the method used for the time integration of the cases presented in this report.

The Newmark explicit form for the displacement evaluation can be written as [11] :

$$\begin{aligned} \mathbf{v}_{i,n+1} = & \left[\frac{\mathbf{M}}{h^2} + \frac{\gamma}{h} \mathbf{D} + \frac{1}{2} \beta \mathbf{K} \right]^{-1} \left[\left(\frac{2\mathbf{M}}{h^2} + (2\gamma - 1) \mathbf{D} + (\beta - \gamma - 0.5) \mathbf{K} \right) \mathbf{v}_{i,n} \right. \\ & \left. + \left(-\frac{\mathbf{M}}{h^2} - ((\gamma - 1)/h) \mathbf{D} + (0.5\beta - \gamma + 0.5) \mathbf{K} \right) \mathbf{v}_{i,n-1} + \mathbf{F}_n \right] \end{aligned} \quad (2.1)$$

where $\mathbf{v}_{i,n}$ is the displacement of node i at the time n . The time step is denoted h , \mathbf{M} is the mass matrix, \mathbf{D} the damping matrix, \mathbf{K} the stiffness matrix and \mathbf{F} contains the forces applied on the nodes.

The Newmark method depends on the numerical parameters β and γ . As Figure 2.1 shows, these two parameters are determining the stability of the solution. For a value of $\gamma < 0.5$ the method is unstable whatever the value of β . If β is greater or equal to $0.25 \cdot (0.5 + \gamma)^2$, the method is unconditionally stable, reminding that the γ has to be greater or equal to 0.5. In the case of β lower than $0.25 \cdot (0.5 + \gamma)^2$ the method becomes conditionally stable. The condition to satisfy is $(\gamma + 0.5)^2 - 4\beta \leq 4/(\omega^2 h^2)$. The various method developed with different combinations of the parameters can be found in Table 2.1.

Each of these methods possess their advantages and drawbacks. For instance, the purely explicit algorithm is more a theoretical method since it is not practically used due to its unstable condition (see Figure 2.1). The average constant acceleration algorithm corresponds to the exact case of an unstable Newmark algorithm. There are no numerical damping associated and the error is of the second order. However, the method used most of the time when it comes to impacts, contact and/or explosions, is the central difference due to its second-order accuracy and simplicity [26]. Like the average constant acceleration algorithm, there is no numerical damping and the error is of the second order but the stability limit is

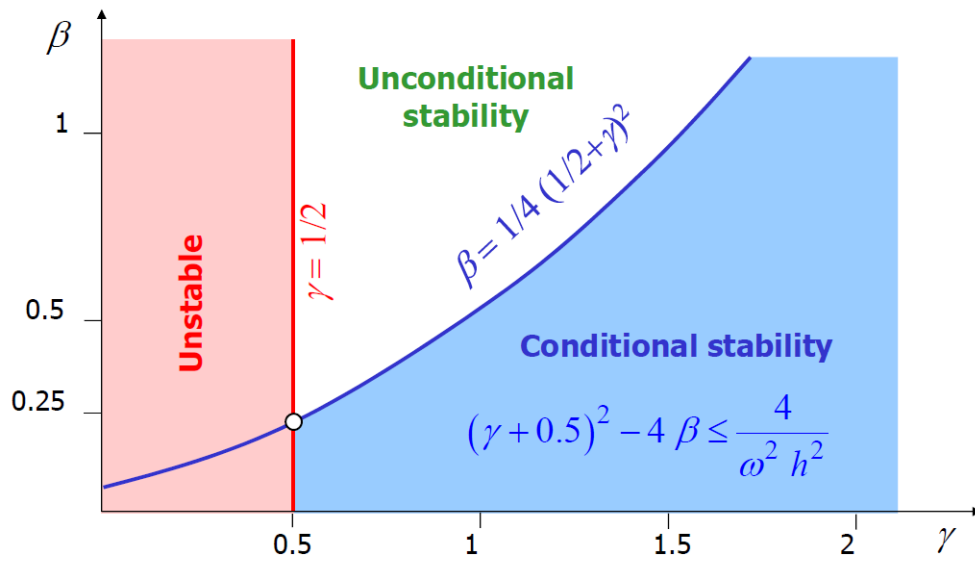


Fig. 2.1 Newmark method stability regarding the parameters β and γ [12].

Algorithm	γ	β	Stability limit ωh	Numerical damping $\bar{\varepsilon}$	Periodicity error $\frac{\Delta T}{T}$
Purely explicit	0	0	0	$\frac{-\omega h}{4}$	-
Central difference	$\frac{1}{2}$	0	2	0	$\frac{-\omega^2 h^2}{24}$
Fox & Goodwin	$\frac{1}{2}$	$\frac{1}{12}$	2.45	0	$O(\omega^4 h^4)$
Linear acceleration	$\frac{1}{2}$	$\frac{1}{6}$	3.46	0	$\frac{\omega^2 h^2}{12}$
Average constant acceleration	$\frac{1}{2}$	$\frac{1}{4}$	∞	0	$\frac{\omega^2 h^2}{12}$

Table 2.1 Newmark methods family.[15]

only of 2 while it was infinite for the other method. This stability limit depends on the time step, h . A discussion about the impact of the time step on the results is carried on in Chapter 3.

Replacing the parameters β and γ by the values corresponding to the central difference, $\beta = 0$ and $\gamma = 0.5$, Equation (2.1) becomes:

$$\mathbf{v}_{i,n+1} = \left[\frac{\mathbf{M}}{h^2} + \frac{1}{2h} \mathbf{D} \right]^{-1} \left[\left(\frac{2\mathbf{M}}{h^2} - \mathbf{K} \right) \mathbf{v}_{i,n} + \left(-\frac{\mathbf{M}}{h^2} + \frac{1}{2h} \mathbf{D} \right) \mathbf{v}_{i,n-1} + \mathbf{F}_n \right] \quad (2.2)$$

2.3 Reduction techniques

2.3.1 Principle

As mentioned in the introduction, finite element model can be built to represent large and/or complex structures. In order to accurately simulate the physical behaviours of those structures, the model requires a significant amount of elements, which is leading to a heavy computation cost and significant time associated to those computations. A solution to reduce the numerical costs is to reduce the domain on which the analysis is performed, in other words, reduce the amount of degrees of freedom studied. This is the precise goal of the reduction method that is presented in this section.

Considering a structure defined by a model containing p degrees of freedom, this can be described by a $[p \times p]$ system represented by the equation:

$$\mathbf{K}\mathbf{x} - \omega^2 \mathbf{M}\mathbf{x} = 0 \quad (2.3)$$

where \mathbf{K} is the stiffness matrix, \mathbf{M} the mass matrix, ω the eigen-frequencies and \mathbf{x} the displacement degrees of freedom of the nodes.

One can reduce it to the model of smaller size $[m \times m]$, with m smaller than p , with \mathbf{y} degrees of freedom retained. This reduced model can be described by:

$$\bar{\mathbf{K}}\mathbf{y} - \omega^2 \bar{\mathbf{M}}\mathbf{y} = 0 \quad (2.4)$$

where the notation $\bar{\mathbf{K}}$ and $\bar{\mathbf{M}}$ corresponds to the reduced structural matrices. The relation between \mathbf{x} and \mathbf{y} is given by:

$$\mathbf{x} = \mathbf{R}\mathbf{y} \quad (2.5)$$

and the two relations between the reduced and the full system are:

$$\bar{\mathbf{K}} = \mathbf{R}^T \mathbf{K} \mathbf{R} \quad (2.6)$$

$$\bar{\mathbf{M}} = \mathbf{R}^T \mathbf{M} \mathbf{R} \quad (2.7)$$

where the matrix \mathbf{R} is the matrix allowing the transition from one state to the other. It depends on the reduction method considered. The focus, in the scope of this project, is put on the Craig-Bampton's method in particular knowing that other methods are existing. For instance, the Guyan-Iron method [14], despite the fact that the method is generally considered as being simple, it offers the basis of the reduction process and even lead to the construction of the Craig-Bampton's method [9].

2.3.2 Guyan-Iron method [14]

This method is based on the condensation of unwanted degrees of freedom (d.o.f.). Considering that one desires to eliminate the \mathbf{x}_C d.o.f. and keep the \mathbf{x}_R with $\mathbf{x}_C + \mathbf{x}_R = \mathbf{x}$ ones, it is possible to write both the mass and stiffness matrices in block matrices such as:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_R \\ \mathbf{x}_C \end{pmatrix}, \mathbf{K} = \begin{pmatrix} \mathbf{K}_{RR} & \mathbf{K}_{RC} \\ \mathbf{K}_{CR} & \mathbf{K}_{CC} \end{pmatrix} \text{ and } \mathbf{M} = \begin{pmatrix} \mathbf{M}_{RR} & \mathbf{M}_{RC} \\ \mathbf{M}_{CR} & \mathbf{M}_{CC} \end{pmatrix} \quad (2.8)$$

The condensed degrees of freedom, \mathbf{x}_C , are composed of both static, \mathbf{x}_S , and dynamic contributions, \mathbf{x}_D [13], such as:

$$\mathbf{x}_C = \mathbf{x}_S + \mathbf{x}_D \quad (2.9)$$

where the static contribution is directly related to the retained d.o.f. by:

$$\mathbf{x}_S = -\mathbf{K}_{CC}^{-1} \mathbf{K}_{CR} \mathbf{x}_R \quad (2.10)$$

The Guyan-Iron method consists in doing the strong assumption of neglecting the dynamic contribution which allows to directly write \mathbf{R} as equation (2.11) where \mathbf{I} is the identity matrix.

$$\mathbf{R}_{GI} = \begin{pmatrix} \mathbf{I} \\ -\mathbf{K}_{CC}^{-1} \mathbf{K}_{CR} \end{pmatrix} \quad (2.11)$$

While this method as proven to be effective in many cases by its simplicity and accuracy for the first modes of vibration, it can be improved by taking into account the dynamic part that has been neglected so far as indicated above. This is why a adapted and more pertinent approach have been proposed by Craig and Bampton [9].

2.3.3 Craig-Bampton method [9]

This dynamic contribution is seen as a interaction of the eigen-modes of the parts of the structure that have been condensed. The amount of constrained modes considered depends on the degree of accuracy wanted, keeping in mind that the more modes the larger the matrices and the computational cost associated to the calculation.

The \mathbf{R} matrix can be expressed as:

$$\mathbf{R}_{CB} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{K}_{CC}^{-1}\mathbf{K}_{CR} & \mathbf{\Phi}_r \end{pmatrix} \quad (2.12)$$

where \mathbf{I} stands for the identity matrix, and $\mathbf{\Phi}_r$ is a diagonal matrix containing the r eigen-modes of d.o.f.. It is obtained by solving the eigen value problem of the condensed part:

$$\mathbf{K}_{CC}\mathbf{x}_C - \omega'^2\mathbf{M}_{CC}\mathbf{x}_C = 0 \quad (2.13)$$

One can notice that if the amount of eigen-modes, r , is set to zero, the Craig-Bampton method is equivalent to the Guyan-Iron method.

The analysis of the dependence of the results with the reduction parameter is addressed in Section 4.2.2.

2.4 Surface interpolation procedures

The finite element model built can not exactly be used as it is for the modelling of the impacted surface. The linearity of the connection between the nodes of the discretized mesh induces a discontinuity of the normal to the surface. This may lead to numerical discontinuity and numerical instabilities resulting in wrong or suspicious results. In order to not heavily increase the computation process with quadratic meshes or iso-geometric analysis to counter that linear connection issue induced by the mesh, an interpolating smooth surface can be built to numerically link the nodes [3].

The construction of a smooth and continuous surface passing through all the nodes of the finite element model can be achieved thanks to the use of interpolating splines, and more generally b-splines [18].

The evaluation of those b-splines in a fast and numerically stable way as been proposed by Carl R. de Boor [6].

2.4.1 Cox-de Boor formula

In the most general way, a b-spline $S(u)$ is defined as :

$$S(u) = \sum_{i=1}^n \mathbf{Q}_i B_{i,d}(u) \quad (2.14)$$

where the \mathbf{Q} are known as n control points and $B_{i,d}(u)$ are b-spline functions of degree d determined thanks to the Cox-de Boor algorithm via:

$$B_{i,0}(u) = \begin{cases} 1 & \text{if } U_i \leq u < U_{i+1} \\ 0 & \text{otherwise.} \end{cases} \quad (2.15)$$

$$B_{i,d}(u) = \begin{cases} \frac{u-U_i}{U_{i+d}-U_i} B_{i,d-1}(u) + \frac{U_{i+d+1}-u}{U_{i+d+1}-U_{i+1}} B_{i+1,d-1}(u) & \text{if } U_i \leq u < U_{i+1} \\ 0 & \text{otherwise.} \end{cases} \quad (2.16)$$

which means that the parameter u has to be comprise between the two values, U_i, U_{i+1} , of the node vector \mathbf{U} , for the polynomial functions $B_{i,d}(u)$ to be non-zero.

2.4.2 Bicubic b-spline surfaces

Since it is not just a simple b-spline that has to be built in order to interpolate the finite element model but a whole b-spline surface, one can adapt equation (2.14) into:

$$S(u, v) = \sum_{i=1}^n \sum_{j=1}^m \mathbf{Q}_{i,j} B_{i,p}(u) B_{j,d}(v) \quad (2.17)$$

The goal being the construction of a surface with a continuous slope and curvature in order to avoid irregularities in the behaviour of the normal, the b-spline surface is set to be a bi-cubic b-spline ($d = p = 3$).

The building of a bi-cubic b-spline requires the existence of a matrix ϕ that allows the transformation of the space such as the points describing the surface can be represented as evenly spaced in a rectangular domain. It allows the construction of matrices for each

coordinate such as the matrix of passing points \mathbf{P} becomes \mathbf{P}_x , \mathbf{P}_y and \mathbf{P}_z of dimension $(n - p + 1) \times (m - d + 1)$. This transformation is represented in Figure 2.2.

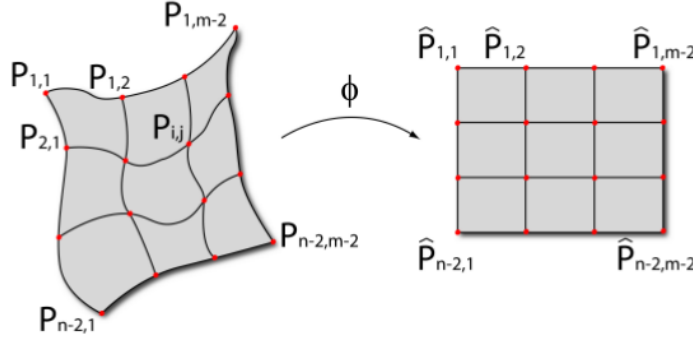


Fig. 2.2 Transformation into a rectangular domain extracted from [3].

As said in the formalism of the Cox-de Boor algorithm, the matrix \mathbf{Q} contains the information relative to the control points of the surfaces. They can be transformed exactly like the passing points \mathbf{P} , which gives the $(n \times m)$ matrices \mathbf{Q}_x , \mathbf{Q}_y and \mathbf{Q}_z .

The two parameters u and v are both used to define the position on the surface using patches, a patch being delimited by 4 points. u and v are defining uniform node vector such as $\mathbf{U} = [1, 2, 3, \dots, n + p + 1]$ for the node vector associated to u and $\mathbf{V} = [1, 2, 3, \dots, m + d + 1]$ for the one associated to v . As previously established, the degrees of a bi-cubic b-spline surface are both set to 3.

Replacing the value of d and p directly in equation (2.16) leads to:

$$\begin{aligned}
 B_{i,3}(u) = & \frac{u - U_i}{U_{i+3} - U_i} \left[\frac{u - U_i}{U_{i+2} - U_i} \left(\frac{u - U_i}{U_{i+1} - U_i} B_{i,0}(u) + \frac{U_{i+2} - u}{U_{i+2} - U_{i+1}} B_{i+1,0}(u) \right) \right. \\
 & + \frac{U_{i+3} - u}{U_{i+3} - U_{i+1}} \left(\frac{u - U_{i+1}}{U_{i+2} - U_{i+1}} B_{i+1,0}(u) + \frac{U_{i+3} - u}{U_{i+3} - U_{i+2}} B_{i+2,0}(u) \right) \Big] \\
 & + \frac{U_{i+4} - u}{U_{i+4} - U_{i+1}} \left[\frac{u - U_{i+1}}{U_{i+3} - U_{i+1}} \left(\frac{u - U_{i+1}}{U_{i+2} - U_{i+1}} B_{i+1,0}(u) + \frac{U_{i+3} - u}{U_{i+3} - U_{i+2}} B_{i+2,0}(u) \right) \right. \\
 & + \frac{U_{i+4} - u}{U_{i+4} - U_{i+2}} \left(\frac{u - U_{i+2}}{U_{i+3} - U_{i+2}} B_{i+2,0}(u) + \frac{U_{i+4} - u}{U_{i+4} - U_{i+3}} B_{i+3,0}(u) \right) \Big]
 \end{aligned} \tag{2.18}$$

an equivalent equation for v can be written:

$$\begin{aligned}
 B_{j,3}(v) = & \frac{v - V_j}{V_{j+3} - V_j} \left[\frac{v - V_j}{V_{j+2} - V_j} \left(\frac{v - V_j}{V_{j+1} - V_j} B_{j,0}(v) + \frac{V_{j+2} - v}{V_{j+2} - V_{j+1}} B_{j+1,0}(v) \right) \right. \\
 & + \frac{V_{j+3} - v}{V_{j+3} - U_{j+1}} \left(\frac{v - U_{j+1}}{V_{j+2} - V_{j+1}} B_{j+1,0}(v) + \frac{V_{j+3} - v}{V_{j+3} - V_{j+2}} B_{j+2,0}(v) \right) \left. \right] \\
 & + \frac{V_{j+4} - v}{V_{j+4} - V_{j+1}} \left[\frac{v - V_{j+1}}{V_{j+3} - V_{j+1}} \left(\frac{v - V_{j+1}}{V_{j+2} - V_{j+1}} B_{j+1,0}(v) + \frac{V_{j+3} - v}{V_{j+3} - V_{j+2}} B_{j+2,0}(v) \right) \right. \\
 & + \frac{V_{j+4} - v}{V_{j+4} - V_{j+2}} \left(\frac{v - V_{j+2}}{V_{j+3} - V_{j+2}} B_{j+2,0}(v) + \frac{V_{j+4} - v}{V_{j+4} - V_{j+3}} B_{j+3,0}(v) \right) \left. \right]
 \end{aligned} \tag{2.19}$$

Using equation (2.15), one may deduce that non-zero values are obtained for the following specific values of i and j :

$$i = \begin{cases} r-3 \\ r-2 \\ r-1 \\ r \end{cases} \quad \text{and } j = \begin{cases} h-3 \\ h-2 \\ h-1 \\ h \end{cases} \tag{2.20}$$

Reintroducing these results in equation (2.17), one obtains:

$$\begin{aligned}
 \mathbf{S}(u, v) = & B_{h-3,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-3} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-3} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-3} + B_{r,3}(u) \mathbf{Q}_{r,h-3} \right] \\
 & + B_{h-2,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-2} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-2} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-2} + B_{r,3}(u) \mathbf{Q}_{r,h-2} \right] \\
 & + B_{h-1,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-1} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-1} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-1} + B_{r,3}(u) \mathbf{Q}_{r,h-1} \right] \\
 & + B_{h,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h} + B_{r,3}(u) \mathbf{Q}_{r,h} \right]
 \end{aligned} \tag{2.21}$$

Re-writing equation (2.21) as third degree polynomial equation, one can express it as:

$$\mathbf{S}(u, v) = \frac{1}{36} \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{r-3,h-3} & \mathbf{Q}_{r-3,h-2} & \mathbf{Q}_{r-3,h-1} & \mathbf{Q}_{r-3,h} \\ \mathbf{Q}_{r-2,h-3} & \mathbf{Q}_{r-2,h-2} & \mathbf{Q}_{r-2,h-1} & \mathbf{Q}_{r-2,h} \\ \mathbf{Q}_{r-1,h-3} & \mathbf{Q}_{r-1,h-2} & \mathbf{Q}_{r-1,h-1} & \mathbf{Q}_{r-1,h} \\ \mathbf{Q}_{r,h-3} & \mathbf{Q}_{r,h-2} & \mathbf{Q}_{r,h-1} & \mathbf{Q}_{r,h} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \quad (2.22)$$

Knowing the coordinates of the passing points \mathbf{P} , one needs to obtain the control points \mathbf{Q} in order to introduce them in equation 2.22 and to compute the surface at any position u, v . Since the degrees are such that $d = p = 3$, the surface is made of $(n - 2) \times (m - 2)$ passing points and of $(n \times m)$ control points. Therefore, it is possible to link the control and the passing points via a $((n - 2) \cdot (m - 2) \times nm)$ matrix named \mathbf{A} :

$$\mathbf{P} = \mathbf{A}\mathbf{Q} \quad (2.23)$$

Taking into account the boundary conditions, one can finally write:

$$\mathbf{Q} = \mathbf{C}\mathbf{Q}_{int} \quad (2.24)$$

where \mathbf{Q}_{int} are the interior control points, or in other words the ones outside of the boundary of the surface. It means that the matrix \mathbf{Q}_{int} has the same dimension as \mathbf{P} . \mathbf{C} is the matrix separating the interior control points from the exterior ones.

Introducing equation (2.24) in equation (2.23) one can get

$$\mathbf{P} = \mathbf{A}\mathbf{C}\mathbf{Q}_{int} \Leftrightarrow \mathbf{Q}_{int} = [\mathbf{AC}]^{-1}\mathbf{P} \quad (2.25)$$

which, in the end, gives the expression for the control points defining the entire b-spline surface with respect to the passing points.

$$\mathbf{Q} = \mathbf{C}[\mathbf{AC}]^{-1}\mathbf{P} = \mathbf{B}\mathbf{P} \quad (2.26)$$

with the matrix $\mathbf{B} = \mathbf{C}[\mathbf{AC}]^{-1}$.

Given the two matrices \mathbf{A} and \mathbf{C} , all the control points can be obtained and therefore the surface can be built. While the matrix \mathbf{A} can simply be obtained via the construction of patches, the matrix \mathbf{C} depends on whether the surface is open, a plane for instance, or closed in at least one of the directions, like a cylinder.

As previously presented, a patch is defined as a combination of 4 points. For instance, one may consider the four points $\mathbf{P}_{r,h}$, $\mathbf{P}_{r,h+1}$, $\mathbf{P}_{r+1,h}$ and $\mathbf{P}_{r+1,h+1}$ for which one can write the following 4 relations

$$\mathbf{P}_{r,h} = \frac{1}{36}(\mathbf{Q}_{r,h} + 4\mathbf{Q}_{r,h+1} + \mathbf{Q}_{r,h+2} + 4\mathbf{Q}_{r+1,h} + 16\mathbf{Q}_{r+1,h+1} + 4\mathbf{Q}_{r+1,h+2} + \mathbf{Q}_{r+2,h} + 4\mathbf{Q}_{r+2,h+1} + \mathbf{Q}_{r+2,h+2})$$

$$\mathbf{P}_{r+1,h} = \frac{1}{36}(\mathbf{Q}_{r,h+1} + 4\mathbf{Q}_{r,h+2} + \mathbf{Q}_{r,h+3} + 4\mathbf{Q}_{r+1,h+1} + 16\mathbf{Q}_{r+1,h+2} + 4\mathbf{Q}_{r+1,h+3} + \mathbf{Q}_{r+2,h+1} + 4\mathbf{Q}_{r+2,h+2} + \mathbf{Q}_{r+2,h+3})$$

$$\mathbf{P}_{r,h+1} = \frac{1}{36}(\mathbf{Q}_{r+1,h} + 4\mathbf{Q}_{r+1,h+1} + \mathbf{Q}_{r+1,h+2} + 4\mathbf{Q}_{r+2,h} + 16\mathbf{Q}_{r+2,h+1} + 4\mathbf{Q}_{r+2,h+2} + \mathbf{Q}_{r+3,h} + 4\mathbf{Q}_{r+3,h+1} + \mathbf{Q}_{r+3,h+2})$$

$$\mathbf{P}_{r+1,h+1} = \frac{1}{36}(\mathbf{Q}_{r+1,h+1} + 4\mathbf{Q}_{r+1,h+2} + \mathbf{Q}_{r+1,h+3} + 4\mathbf{Q}_{r+2,h+1} + 16\mathbf{Q}_{r+2,h+2} + 4\mathbf{Q}_{r+2,h+3} + \mathbf{Q}_{r+3,h+1} + 4\mathbf{Q}_{r+3,h+2} + \mathbf{Q}_{r+3,h+3})$$

and eventually this leads to the construction of the matrix \mathbf{A} :

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 & 0 & \cdots \\ \vdots & & & & & & & \ddots & & & & & & & & \vdots & \\ 0 & \cdots & 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 & 0 & \cdots \\ 0 & \cdots & 0 & 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & 4 & 1 & 0 & \cdots & 4 & 16 & 4 & 0 & \cdots & 1 & 4 & 1 \end{bmatrix} \quad (2.27)$$

The last element that must be computed for the construction of the b-spline is the \mathbf{C} matrix. However, this matrix depends on the boundary conditions and the type of structure that is model. The open and closed surfaces must be differentiated.

Open Surface

In the case of an open surface, the condition is that there is a symmetry with regard to the internal control points. For instance, considering that $\mathbf{Q}_{1,1} = 2\mathbf{Q}_{2,2} - \mathbf{Q}_{3,3}$ this induces that $\mathbf{Q}_{n,m} = 2\mathbf{Q}_{n-1,m-1} - \mathbf{Q}_{n-2,m-2}$. Considering all the conditions, one can finally obtain **C**:

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & \ddots & & & & & & \vdots \\ 0 & \dots & 0 & 2 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & \ddots & & & & & & \vdots \\ 0 & \dots & 0 & -1 & 2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & -1 & 2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 2 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 2 \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & -1 & 0 & 0 & \dots & 0 & 2 & 0 & \dots & 0 \\ \vdots & & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 0 & 2 \end{bmatrix}$$

Closed Surface

In the case of a closed surface, it is mandatory to impose the continuity C^1 of the b-spline surface at the connections between the extremities of the domain. In the end, the matrix \mathbf{C} for a closed surface can be written as:

$$\mathbf{C} = \begin{bmatrix}
2 & -1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 & -2 & 1 & 0 & \dots & 0 \\
1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 \\
0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & -1 & 0 & \dots & 0 \\
\vdots & & & & & & & \ddots & & & & & & & \vdots \\
0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & -1 & 0 \\
0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & -1 \\
0 & \dots & 0 & 1 & 2 & 0 & \dots & 0 & -1 & 2 & 0 & \dots & 0 & 1 & -2 \\
2 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & & & & & & & \ddots & & & & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 2 \\
-2 & 1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 \\
-1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\
0 & -1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\
\vdots & & & & & & & \ddots & & & & & & & \vdots \\
0 & \dots & 0 & -1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\
0 & \dots & 0 & 0 & -1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 1 \\
0 & \dots & 0 & 1 & -2 & 0 & \dots & -1 & 2 & 0 & \dots & 0 & 0 & -1 & 2
\end{bmatrix}$$

2.5 Contact treatment

The last theoretical aspect that requires to be introduced here and that represents the core of this project concerns the contact detection and its treatment.

2.5.1 Distance Computation using Newton-Raphson method

While in principle the determination of the distance is a fairly simple task, the computation of the shortest distance between a node and a surface represents a nonlinear algebraic problem [4]. The algorithm corresponding to the determination of that closest point is based by the Newton-Raphson algorithm.

The first step of the algorithm is to determine the closest patch of the spline with respect to the nodes of the impacting body. In order to establish which patch is the closest to each node, the distance with the nodes of the spline, \mathbf{P} , is computed. Knowing the few closest points of the spline to the body, it is possible to determine to which patch those close points are belonging. Whenever three points of the same patch are present in those closest points, the associated patch is then considered to be the closest.

Within those patches, a point $S(u_{n+1}^i, v_{n+1}^i)$ is arbitrarily chosen within the closest patch for each main body node. In order to initiate the algorithm, this first point corresponding to the initial guess is set at the center of the patch. This assumption is only made at the very first time step. Since usually time step for contact simulations is small, it is safe to consider that the closest point of the previous time step is a good and pertinent starting point for the initialization of the iteration at the current time step. The purpose of this assumption helps to speed up the computation, since the closest the guess is from the actual value the fastest the convergence towards it will be. A graphical representation of the whole process of the algorithm is given by Figure 2.4 where Γ_{n+1}^i is one of the nodes of the main structure.

The general flowchart of the Newton-Raphson algorithm is given by Figure 2.3.

An important parameter that appears in the description of the algorithm is the stop criterion ϵ_{NR} . It imposes the precision at which one desires to find the closest point. A discussion about its impact on the solution is carried on in Section 4.2.4.

2.5.2 Contact forces

Whenever, during the simulation, the evaluated distance computed with the Newton-Raphson algorithm becomes negative, it means that there is a penetration between the two bodies. This non physical behaviour must be corrected.

Using the Newmark algorithm presented in the first section of this Chapter, a prediction is made on either one or the two body displacements. Their positions are updated with respect to the positions and the new distance evaluated using the algorithm presented in Figure 2.3. In the case of a positive distance, the time t is increased by the time step h . If there is penetration, the corrections must be computed and added to the prediction.

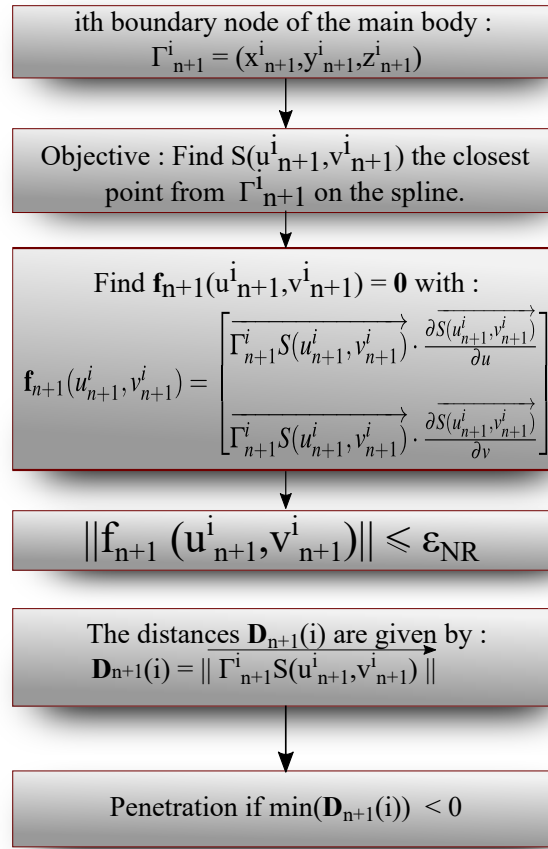


Fig. 2.3 Flowchart of the Newton-Raphson for the distance computation.

Multiple algorithm exist to deal with the contact. In the scope of this project, two are considered. The first one is the Lagrange Multipliers [16], and the second one is the Penalty method [19].

This algorithm is described in the flowchart represented in Figure 2.5.

Lagrange Multiplier

The Lagrange Multiplier method [16] relies on the computation of surface contact forces, $\boldsymbol{\lambda}$. In order to obtain the values of $\boldsymbol{\lambda}$, a contact matrix, named \mathbf{MC} , must be computed. In the case of flexible bodies, they are susceptible to move in response to the contact of one with another, a contact matrix must be computed for each one. The first step consists on the computation of the outer normal vector to the spline surface, $\mathbf{n}_{out,n+1}$ and the tangential vector to that same surface, \mathbf{t}_{n+1} both at the closest points determined with the computation

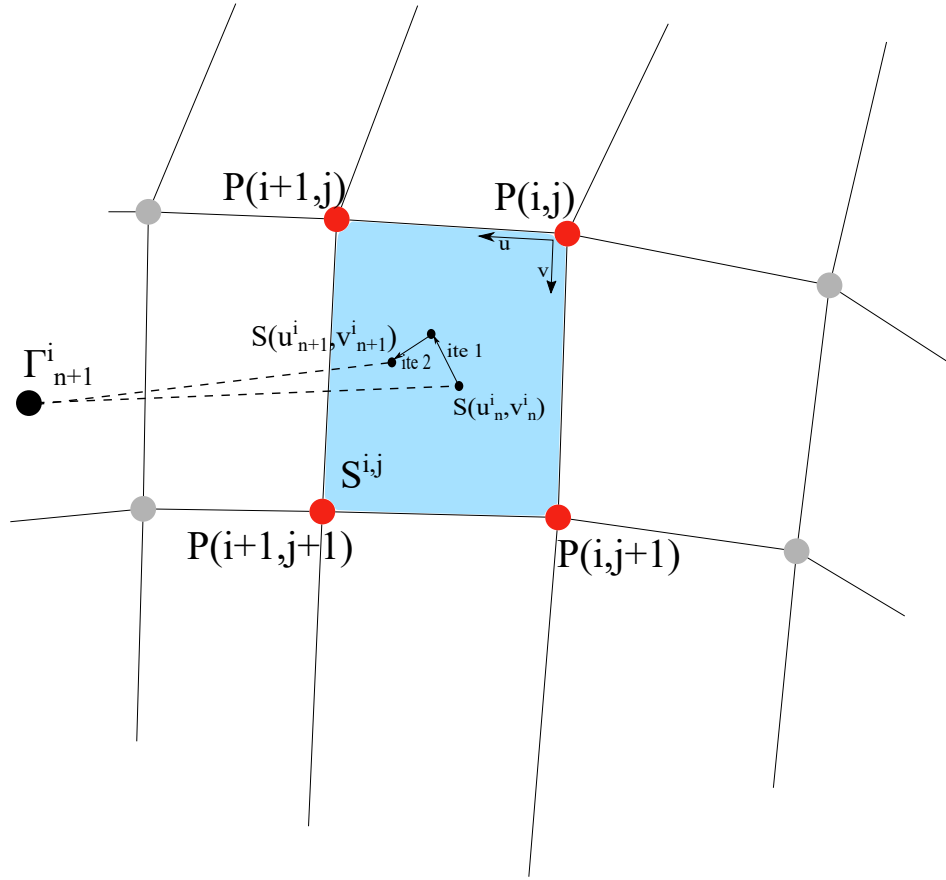


Fig. 2.4 Graphical representation of the Newton-Raphson iterations.

of the distance:

$$\mathbf{n}_{out,n+1}^i = -\frac{\overrightarrow{\partial S(u_{n+1}^i, v_{n+1}^i)}}{\partial u} \wedge \frac{\overrightarrow{\partial S(u_{n+1}^i, v_{n+1}^i)}}{\partial v} \quad (2.28)$$

$$\mathbf{t}_{n+1}^i = \mathbf{n}_{out,n+1}^i \wedge (\mathbf{n}_{out,n+1}^i \wedge \overrightarrow{\Gamma_n^i \Gamma_{n+1}^i}) \quad (2.29)$$

where $\overrightarrow{\Gamma_n^i \Gamma_{n+1}^i}$ corresponds to the vector given by the position of the node i at t_n , Γ_n^i and the position of the prediction at the time t_{n+1} , Γ_{n+1}^i .

A graphical representation of the values to compute is shown in Figure 2.6.

The next step is related to the determination of the contact constraint matrices \mathbf{VC}_{n+1} and \mathbf{VCT}_{n+1} . The matrix \mathbf{VCT}_{n+1} depends on the friction coefficient, whenever it is equal

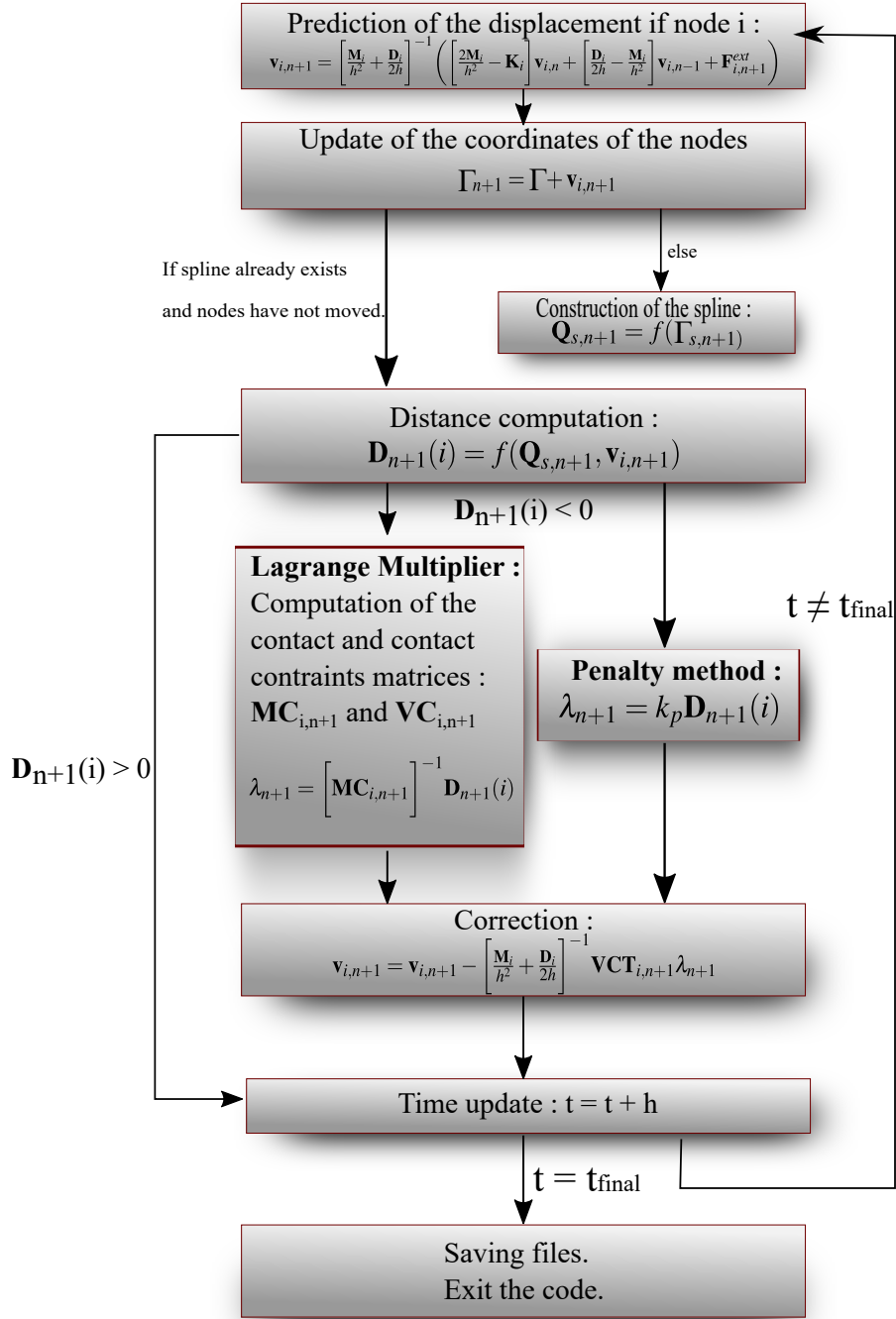


Fig. 2.5 Flowchart of the contact treatment.

to zero, the matrix is identical to \mathbf{VC}_{n+1} :

$$\mathbf{VC}_{n+1}^i = \hat{\mathbf{B}} \cdot \mathbf{c} \cdot \mathbf{n}_{out,n+1}^i \quad (2.30)$$

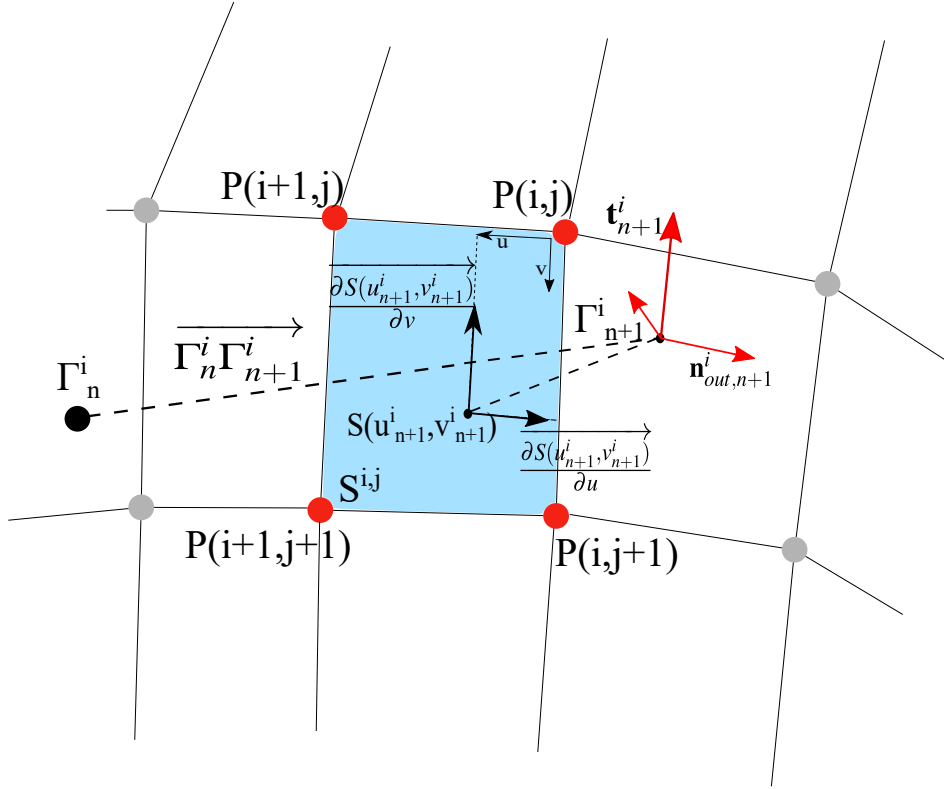


Fig. 2.6 Graphical representation of the penetration of a node and its correction.

$$\mathbf{VCT}_{n+1}^i = \hat{\mathbf{B}} \cdot \mathbf{c} \cdot (\mathbf{n}_{out,n+1}^i + \mu \mathbf{t}_{out,n+1}^i) \quad (2.31)$$

The matrix $\hat{\mathbf{B}}$ contains the information about the current patch, $S^{i,j}$. It corresponds to a restriction on only one patch per node of the matrix that has been described in equation (2.26).

Finally, the matrices can be computed such as:

$$\mathbf{MC}_{n+1} = \mathbf{VC}_{n+1}^T \left[\frac{\mathbf{M}}{h^2} + \frac{\mathbf{D}}{2h} \right]^{-1} \mathbf{VCT}_{n+1} \quad (2.32)$$

Let's mention that depending on the contact scenario, for instance in a very irregular configuration, this matrix has to be lumped, which means that the sum of each row is placed on the diagonal of the matrix while the rest of it is set to zero. By this process, it helps to stabilize the solution.

The contact forces, λ , can now be evaluated for the nodes in contact with the spline.

$$\lambda_{n+1} = \left[\mathbf{MC}_{b1,n+1} + \mathbf{MC}_{b2,n+1} \right]^{-1} \mathbf{d}_{n+1}^* \quad (2.33)$$

If only one of the bodies is flexible, the matrix $\mathbf{MC}_{b2,n+1}$ is equal to 0.

Using the contact forces, the nodes that were penetrating the surface represented by the spline have their displacement corrected. The negative displacement becomes equal to zero leaving the two bodies in a perfect contact.

Penalty method

While for the Lagrange Multiplier the solution of equation 2.33 is based on the computation of contact matrices, which involves heavy matrix calculation such as the calculation of the inverse of the contact matrices, the Penalty method set λ such as:

$$\lambda_{n+1} = k_p \mathbf{d}_{n+1}^* \quad (2.34)$$

with k_p is a scalar parameter defined by the user before the computation. Unfortunately, the simplicity of this statement comes with drawbacks, such as the fact that the penetration of the bodies is allowed and the strong dependence of the final results to the penalty parameter k_p . The differences between the two methods are graphically represented in Figure 2.7. A detailed discussion about this important aspect is carried on in Chapter 4.

In the end, once the contact forces, λ can be evaluated, the correction can be computed such as:

$$\mathbf{v}_{i,n+1}^{cor} = \left[\frac{\mathbf{M}_i}{h^2} + \frac{\mathbf{D}_i}{2h} \right]^{-1} \mathbf{VCT}_{i,n+1} \lambda_{n+1} \quad (2.35)$$

and the final displacement is equal to :

$$\mathbf{v}_{i,n+1} = \mathbf{v}_{i,n+1} - \mathbf{v}_{i,n+1}^{cor} \quad (2.36)$$

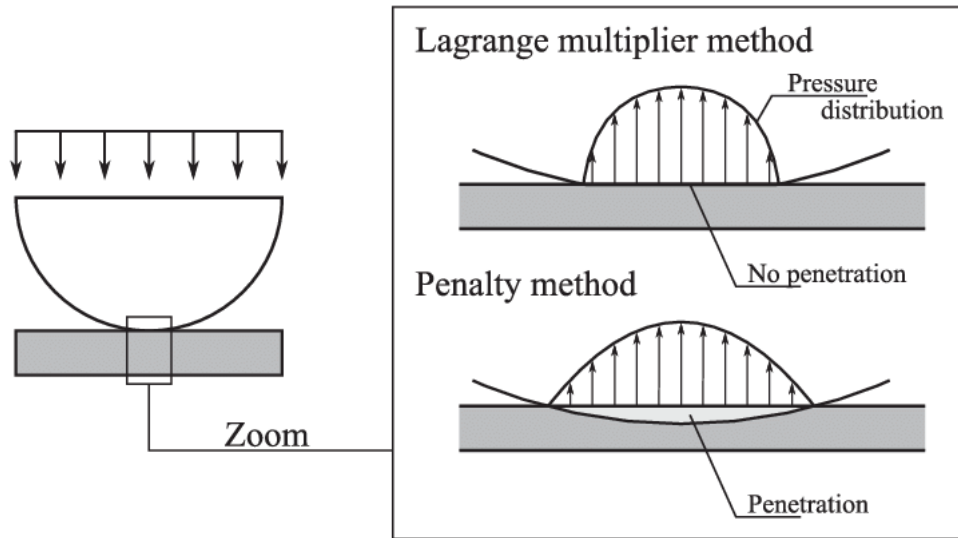


Fig. 2.7 Graphical comparison of Lagrange Multiplier method and Penalty method [7].

2.6 Conclusions

This chapter provides a global presentation of the various algorithms considered in this project. The time integration methods of the Newmark's family are presented and in particular the central difference which is the one of interest in the scope of this project. An introduction to the reduction methods and in particular the Craig-Bampton method has then been given. While with the theoretical presentation of reduction method is not easy to perceive, a practical analysis is performed in Chapter 4. In the third section, the equations used for the computation of the b-spline have been presented. The interpolation of the nodes of the mesh with a certain continuity of the b-spline enforced the consideration of a cubic b-spline surface. The cost of such a surface is that the computation of the exact distance between the two bodies requires the usage of the Newton-Raphson algorithm. This approach consist in solving a nonlinear algebraic problem with the algorithm provided in this section. Finally, the two methods to model the contact (i.e. the Langrange Multipliers and the Penalty method) have been theoretically presented. A practical comparison of the two is given in Chapter 4 and 5. The next chapter, provides a practical description of the numerical tool that contains all the algorithms described in this chapter.

Chapter 3

Notice and code presentation

Introduction

This chapter is dedicated to the presentation of the numerical tool that has been developed in the scope of this project on a practical point of view. The open-source aspect of the tool is explained. The usage of multiple programming languages and the communication between them are presented. The initialization file of the code is also described.

In the first section, the building of the finite element model as well as the software that generates those models are introduced. This step is mandatory to extract the exact matrices of mass and stiffness in order to simulate physical cases in the developed tool.

The second section concerns the first step of the developed software which is the reading of the entry file and the application of the reduction method presented in Section 2.3. This has been coded in Python 2.7. It is also about the communication with the core of the code implemented in C++.

The last section deal with the C++ part. The time integration and the contact interactions have been coded in this programming language. The section also contain the explanations about the connections between the different algorithms presented in the Chapter 2. The reason why the tool has been separately coded using two programming languages is also provided.

3.1 Code_Aster & Finite Element Model

As presented in the introduction, the main goal of this project was the creation of a numerical tool to model and conduct a deep analysis of the contact interactions. The secondary goal was that this tool had to be open-source and free from any commercial constraint. The latter statement means that the software is more modular but its general aspect conduces it to be outperformed by another tool developed for one specific task.

In order to keep this open aspect from the very start of the project the decision of an open source software for the construction of finite element model has been taken. The ideal software that corresponds to those criteria is *Code_Aster*.

Code_Aster is a french software developed by E.D.F. ("Électricité de France"). It includes a CAD tool called Salome that is directly linked to the mesh creation tool allowing simple construction. *Code_Aster* can be compared to *Samcef*, developed by Samtech, in the sense that it offers multi-physical analysis and modelling method. Acoustic, thermal and mechanics simulations can be solved thanks to *Code_Aster* [10].



Fig. 3.1 Logo code_aster

The mechanical analysis of *code_aster* allows the application of boundary conditions, creation of materials, construction of associated mass and stiffness matrices and computation of eigen-modes. In addition to that, *code_aster* has been coded in Python. It makes the manipulation of the extracted matrices more easy and natural in a code developed in Python.

3.2 Python code: reduction and communication

Since the extracted matrices from *code_aster* have Python structure, that Python is a user friendly and OS independent language, it is a perfect candidate for the starting point of the numerical tool. On top of that, the input file can directly be interpreted by Python.

The input files takes the following form:

InitCode.txt

```

RESULTS = {

    'Results': '../Code/Results/Simulation01'    # Results save location
}

FILES = {

    'ImpactedSurf': 'ISNodes.dat',            # Impacted surface mesh
    'ImpactedSurfBC': 'ISBC.dat',
    # Boundary conditions of the impacted surface
    'ImpactedSurfOrder': 'ISNodesOrder.dat',
    # Order of the nodes of impacted surface
    'ImpactedSurfDof': 'SDof.txt',            # D.o.f of the impacted surface
    'ImpactedSurfStiff': 'ISStiff.mat',
    # Stiffness matrix of the impacted surface
    'ImpactedSurfMass': 'ISMass.mat',         # Mass matrix of the impacted surface
    'Body': 'Body.dat',                      # Main body mesh
    'BodyBC': 'BBC.dat',                     # Boundary conditions of the main body
    'BodyRed': 'BRed.dat',                   # Retained nodes of the main body
    'BodyDof': 'BDof.txt',                   # D.o.f of the main body
    'BodyStiff': 'BStiff.mat',               # Stiffness matrix of the main body
    'BodyMass': 'BMass.mat',                 # Mass matrix of the main body
}

VARIABLES = {

    'BConstrained': 10,                      # Constrained modes of the main body
    'ISConstrained': 10,                     # Constrained modes of the impacted surface
    'n': 61,                                 # Row in the spline
    'm': 7,                                  # Column in the spline
    'crit_NR': 1e-6,                         # Newton-Raphson parameter
    'h': 1e-7,                               # Time step

```

```

't':5e-1,          # Time of the analysis
'mu':0.0,          # Friction coefficient
'Vx':0.0,          # Body initial velocity along X
'Vy':0.0,          # Body initial velocity along Y
'Vz':0.0,          # Body initial velocity along Z
'Fx':0.0,          # Body initial force along X
'Fy':0.0,          # Body initial force along Y
'Fz':0.0,          # Body initial force along Z
'VSurfx':0.0,      # Impacted surface initial velocity along X
'VSurfy':0.0,      # Impacted surface initial velocity along Y
'VSurfz':0.0,      # Impacted surface initial velocity along Z
'FSurfx':0.0,      # Impacted surface initial force along X
'FSurfy':0.0,      # Impacted surface initial force along Y
'FSurfz':0.0,      # Impacted surface initial force along Z
'VRot':0.0,        # Body initial rotation speed
'RotAxeX':0.0,     # Angle of rotation X
'RotAxeY':0.0,     # Angle of rotation Y
'RotAxeZ':1.0,     # Angle of rotation Z
'kpen':1e8,         # Penalty coefficient
'resolution':10000, # Time resolution for the results save
}

OPTIONS = {

'Reduction':True,   # Reduction of the main body
'BBC':True,         # Boundary conditions of the main body
'ISBC':True,        # Boundary conditions of the impacted surface
'Damping':'MOD',    # Damping consideration
'ContactTreatment':'LAG', # Contact treatment algorithm
'Flexible':False,   # Flexible surface
'Graphics':True     # Graphical display of the simulation
}
c

```

where the different groups are corresponding to Python dictionaries.

The first one is simply the files where the different results generated at the end of the simulation are saved. The results are generated in three to four files, depending on the type of the study, including the evolution of the distance over time, the evolution of the contact forces over time, the displacement along X, Y and Z of the nodes of the body and eventually

the displacement along X, Y and Z of the nodes of the surface.

The second dictionary is the one gathering the name of all the files extracted from *code_aster*. The first one `ImpactedSurf` corresponds to the nodes of the finite element model constructed in the mesh tool of *code_aster* that are used to create the impacted surface represented by a spline. If some of the nodes of this structure are clamped, those nodes are grouped in the file associated to `ImpactedSurfBC`. To build the spline with regular patches, the nodes have to be properly ordered, the file found under `ImpactedSurfOrder` is re-ordering the nodes obtained via the file `ImpactedSurf`. If some boundary conditions are associated to the body, *code_aster* will create extra degrees of freedom, that have no physical meaning, to simplify the calculation within *code_aster*. However, these extra d.o.f are not necessary in the tool. They are have to be located in the matrices and removed. The file `ImpactedSurfDof` contains all the degrees of freedom of the surface and including the position of the non-physical ones. Finally, the two files under `ImpactedSurfStiff` and `ImpactedSurfMass` are the stiffness and mass matrices.

The files under the items `Body`, `BodyBC`, `BodyDof`, `BodyStiff` and `BodyMass` are basically following the same procedure as the ones for the surface but for the main body. In addition `BodyRed` contains the group of nodes that are eventually retained in the case of a reduction.

The third dictionary gathers all the information about the different variables that could eventually change from one simulation to another. The two firsts, `BConstrained` and `ISConstrained` are corresponding to the amount of constrained eigen-modes used in the Craig-Bampton reduction (see Section 2.3). The parameters `n` and `m` are respectively the amount of nodes on each line and column of the spline. `crit_NR` is simply the stop criterion of the Newton-Raphson algorithm (Section 2.5.1), `h` the time step, `t` the duration of the simulation and `mu` the friction coefficient. All the items `Vx`, `Vy`, `Vz`, `Fx`, `Fy`, `Fz` are the velocity and forces acting on the main body along X, Y and Z , while `VSurfx`, `VSurfy`, `VSurfz`, `FSurfx`, `FSurfy`, `FSurfz` are the corresponding to the same thing but for the surface. If the body is rotating, the velocity at which it does is given by `VRot` while the axis around which it rotates is given by the coordinates (`RotAxeX`, `RotAxeY`, `RotAxeZ`). The two last parameters are concerning the value of the penalty coefficient, `kpen`, and the resolution for the save of the results.

The last dictionary contains the different options that are possible in the tool. Turn some off may lead to the obsolescence of some of the other parameters. For instance is the item

Reduction is set to False instead of True, the files containing the nodes of the reduced model are of no use. Same goes for the options BBC and ISBC that specifies if boundary conditions have been added to the structures. If none of them are true the files with the degree of freedom are useless since only physical d.o.f. are existing in the models. The damping of the system can be set to either MOD for modal damping, PROP for proportional damping or to NONE for zero damping. For the contact treatment, the two algorithms presented in section 2.5.2 are available under the parameters LAG for the Lagrange Multipliers or PEN for the Penalty method. If the surface impacted is flexible, the value of 'Flexible' is set to True otherwise it is set to False.

The final parameter of the dictionary OPTIONS, called Graphics is an option for the graphical representation of the simulation. The tool incorporates a direct integrated link with the open source software Gnuplot [1, 2] allowing the "in real time" graphical display. An example of what can be obtained in the tool is represented in Figure 3.2(b).

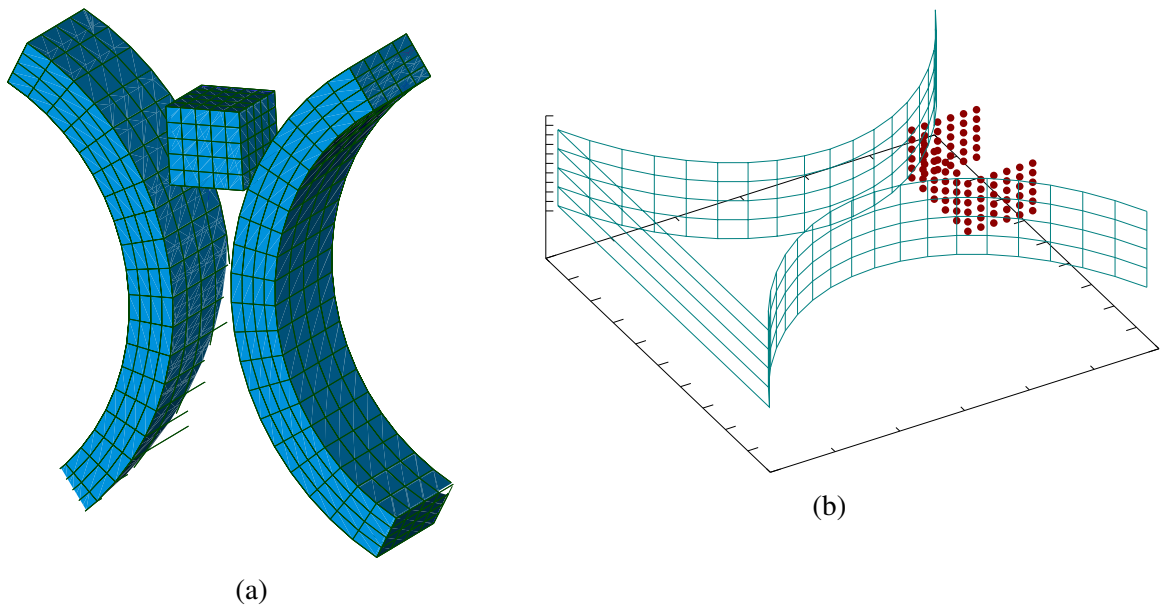


Fig. 3.2 Typical example representing a cube interacting with two rings as a finite element model (a) and the equivalent representation in the tool (b).

The red dots are the one read in the file containing the number of the nodes of the body retained for the reduction and the green lines are drawing the b-spline surface that interpolates

the nodes found under the `ImpactedSurf` item. The equivalent finite element model created in *code_aster* is represented in Figure 3.2(a).

The matrices can then be reduced with the Craig-Bampton method, and transmitted to the core of the code, developed in C++, along side with the other parameters just read from the entry file.

3.3 C++ code: core of the numerical tool

Contrary to Python, which is an interpreted language, C++ is a compiled language. The main difference lay in the process leading to the execution of the code. An interpreted language is first parsed then interpreted and finally executed, and that at every run. On the other hand, a compiled language is compiled once and then can be executed at each run. This results in a significant gain of time in the execution [22, 25].

The tool has been developed with its potential future development. In order to ease the addition of options and possibilities the code have been written in an object oriented aspect [5]. This has the advantage of creating tools for the object that do not require to see the code to understand how to use them. It also follows the "divide to conquer" principle, where big blocks of code containing a lot of actions are turned into small individual tasks.

Concerning the code itself, once the matrices and the different variables, transmitted by the Python code, are received, the surface is built following the algorithm presented in Section 2.4. In order to determine if the surface is closed or not, the algorithm simply check if the last sequence of nodes corresponds to the first sequence or not.

The surface created, a first prediction on the displacement of the main body can be made. Using the initial conditions and the Newmark algorithm (Section 2.2) the new positions of the nodes are estimated. If the second structure, interpolated by the spline, is flexible, a prediction is also emitted for the eventual displacement of the nodes. If this displacement is different of zero, a new spline interpolating the new coordinates of the nodes has to be constructed again.

The treatment of the rotation in the developed tool requires to be presented since its application depends on the contact between the two structures. Actually, none of the bodies

are rotating within the tool even though they physically are. It is the contact zone that is rotating. That means that the global displacement is zero, in the "Newmark prediction" sense, before the first contact. Once this contact happens the response is applied at the original position of the nodes. The response computed at the rotated position, is rotated back at the initial position and applied to the nodes that have not actually moved. So technically the bodies are not moving but are responding to the contact like if they were moving. This specificity allows to simply model the rotation of clamped structure such as turbine blades which constitutes one of the system studied in this report.

With the update of the position of nodes, the distance between each nodes and the spline can be evaluated. To do so, the Newton-Raphson algorithm, is solved. As explained in Section 2.5.1, the four closest surface points of each node of the body are grouped together and called "patches". Within those patches, an initial guess on the position of the closest point is made. At the very first time step, this point is placed at the middle of the patch, its position is then updated using an iterative process until its position is stabilized. This point is the starting point for the distance computation of the next time step.

These two last steps are repeated at every time step. Whenever the evaluated distance is negative, that means that there is a contact between the two bodies. Using either the Penalty method or the Lagrange Multipliers, a correction is computed and added to the prediction.

The global structure of the code as been graphically represented in Figure 3.3.

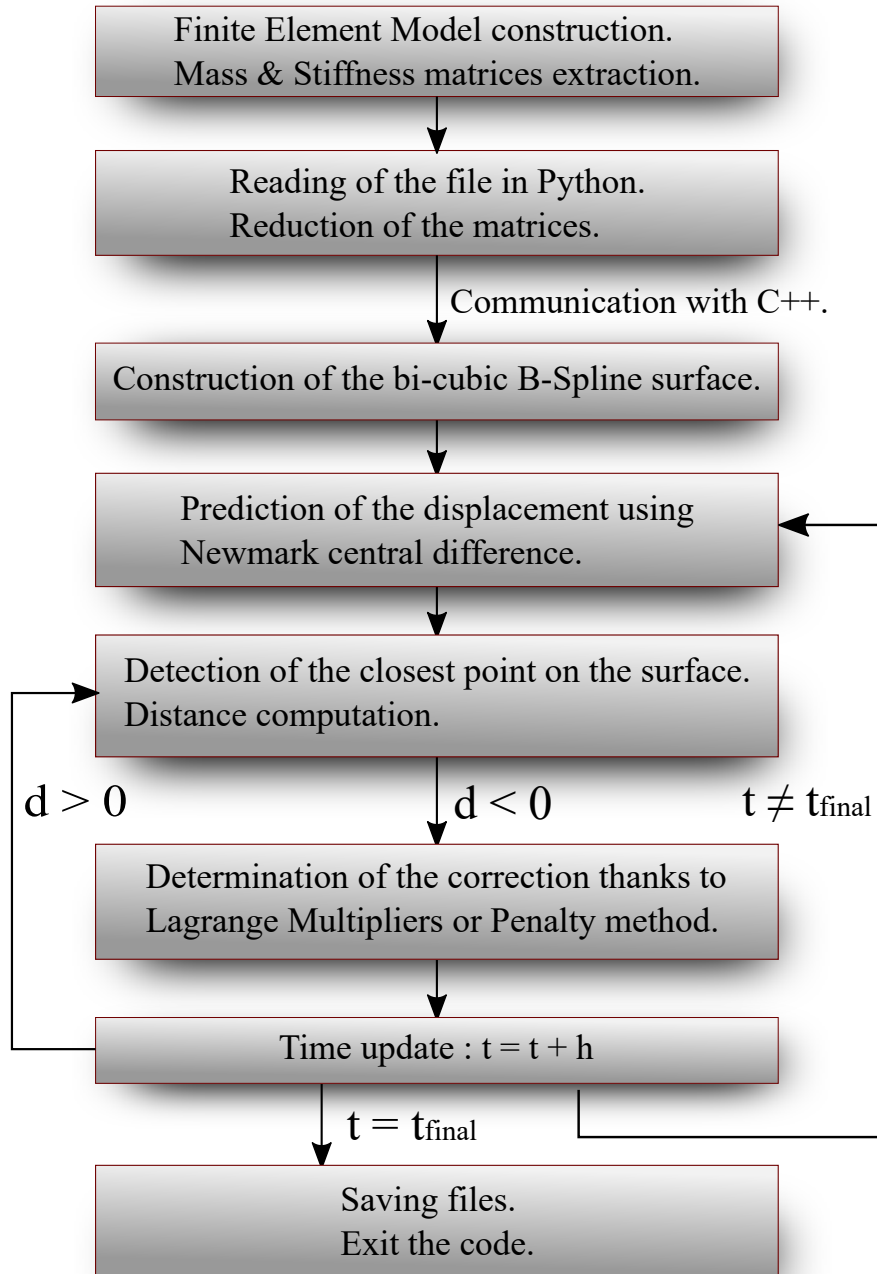


Fig. 3.3 Structure of the code.

Conclusion

In this chapter the overall structure of the code has been presented. Its different aspects and specificities have been demonstrated and explained.

First, the extraction of the mass and stiffness matrices and the construction of a finite element model have been made possible thanks to *Code_Aster*. The reasons that oriented the decision to use such a software have been presented.

Second, the file read at by the developed tool and containing all the useful inputs have been described. Each parameter have been presented in details. The graphical rendering that has been made possible thanks to Gnuplot is exposed as well. The reduction of the extracted matrices is achieved in the code segment presented in this section.

Finally, the core of the code have been presented with the reason that led to switch of programming language. The global structure of the code is given in a flowchart describing the overall behaviour of the code.

Now that the tool has been presented, the next step concerns the validation of the results it provides. This is achieved thanks to the multiple analysis realized on one test case, this topic concerns the next chapter.

Chapter 4

Validation of the numerical tool

Introduction

The topic of this chapter concerns the numerical validation of the developed tool, including the detailed analysis of the convergence of the results, the physical representation of a symmetrical test case and finally the comparison of the two methods to deal with the contact.

The first section is related to the presentation of the test case that is studied throughout the chapter. The global set-up as well as the mechanical properties are also presented.

The second section concerns the convergence analysis of the results regarding the influence of the mesh discretization, the reduction parameter, the time step and the degree of precision of the Newton-Raphson algorithm on the obtained data.

In the third section, an analysis of the results is provided. The dynamic behaviour of the system is described for both rigid and flexible situation. Furthermore, some physical parameters such as the damping and the friction coefficient are introduced and their effects studied.

Finally, in the last section the Lagrange Multipliers method is compared with the Penalty method, for which different factors are considered.

4.1 Presentation of the test case

4.1.1 Geometry and mechanical properties

The studied structure consists in a 5 cm long cube falling at a given velocity, v , between two rings separated, at their closest point, by 4 cm. The rings have a thickness of 3 cm. The outer part correspond to a circular arc of radius equals to 18 cm. This configuration illustrated in Figure 4.1. The red lines at the extremities of the two rings corresponding to clamped parts.

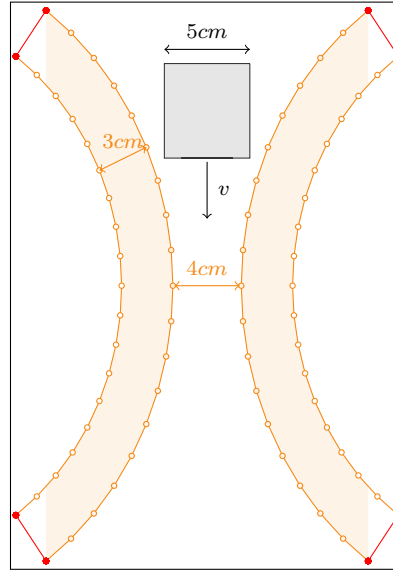


Fig. 4.1 Global configuration of the test case.

The corresponding finite element model, built in *code_aster* is represented in Figure 4.2.

This test case presents some particularities that are interesting for the validation of the developed tool. First, the geometric symmetry allows to verify to well behaviour of the tool with the retrieve of the symmetry in the results. The validation of the symmetry is addressed at the end of this section. Second, the cube being larger than the gap between the rings interesting situations can happen. For instance with enough energy the cube could deform itself enough in order to fit and pass between the rings. Finally, the consideration of friction during the contact can lead the cube to stay stuck between the rings in a permanent deformed state.

Concerning the mechanical properties of the bodies, they are both made of steel with the properties summarized in Table 4.1.

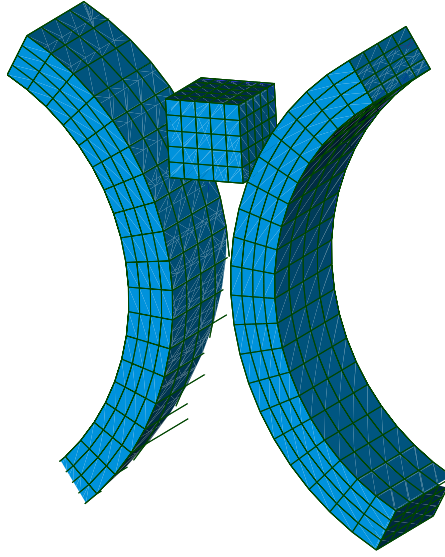


Fig. 4.2 Finite element model of the cube falling between two rings.

Material	Steel
Density	7800 kg/m ³
Poisson's Ratio	0.3
Young Modulus	210 GPa

Table 4.1 Mechanical properties of the cube and the two rings.

4.1.2 Initial Velocity

The speed at which the cube is moving between the rings strongly influences the physical behaviour of the study. Three different cases have been compared, $v = 10$ m/s, $v = 50$ m/s and finally $v = 100$ m/s. The obtained results are represented in Figure 4.3 showing the cube at three different times. For a velocity of 10 m/s Figure 4.3(a), the cube hits the rings but does not have enough energy to deform enough to enter between them. Therefore, after the contact, it slightly deforms before it bounces back. If the cube falls at 100 m/s, Figure 4.3(c), it has enough kinetic energy to deform all the way through the two rings. If now the cube has an intermediate velocity of 50 m/s, Figure 4.3(b), the cube has enough energy to follow the curvature of the rings but not enough to pass, it will instead bounce. This last scenario is the most interesting for us since it contains significant deformations and bouncing back phenomenon. Therefore, throughout this chapter, the velocity of the cube is set at 50 m/s.

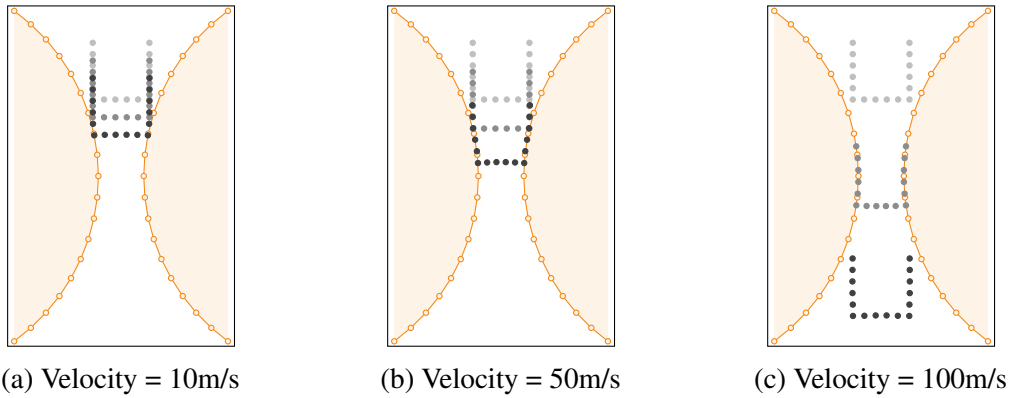


Fig. 4.3 Comparison of different falling velocities for the cube between two rigid rings.

4.1.3 Symmetry of the results

In order to validate the symmetry of the results, the focus is made on two nodes of the bottom face of the cube represented by the red dots in Figure 4.4. Their displacements are compared in Figure 4.5.

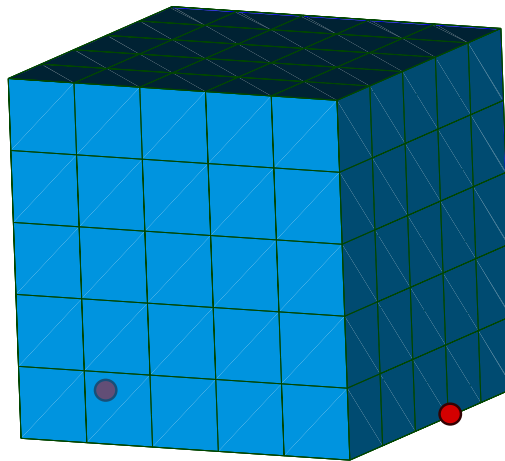


Fig. 4.4 Node of interest on the cube for the validation of the symmetry represented by the red dots.

The symmetry of the geometry of the system studied is well retrieved in the displacement of the nodes of the cube. The displacement along the X and Z axis are exactly opposite while the one along Y is strictly identical for the two nodes (right and left).

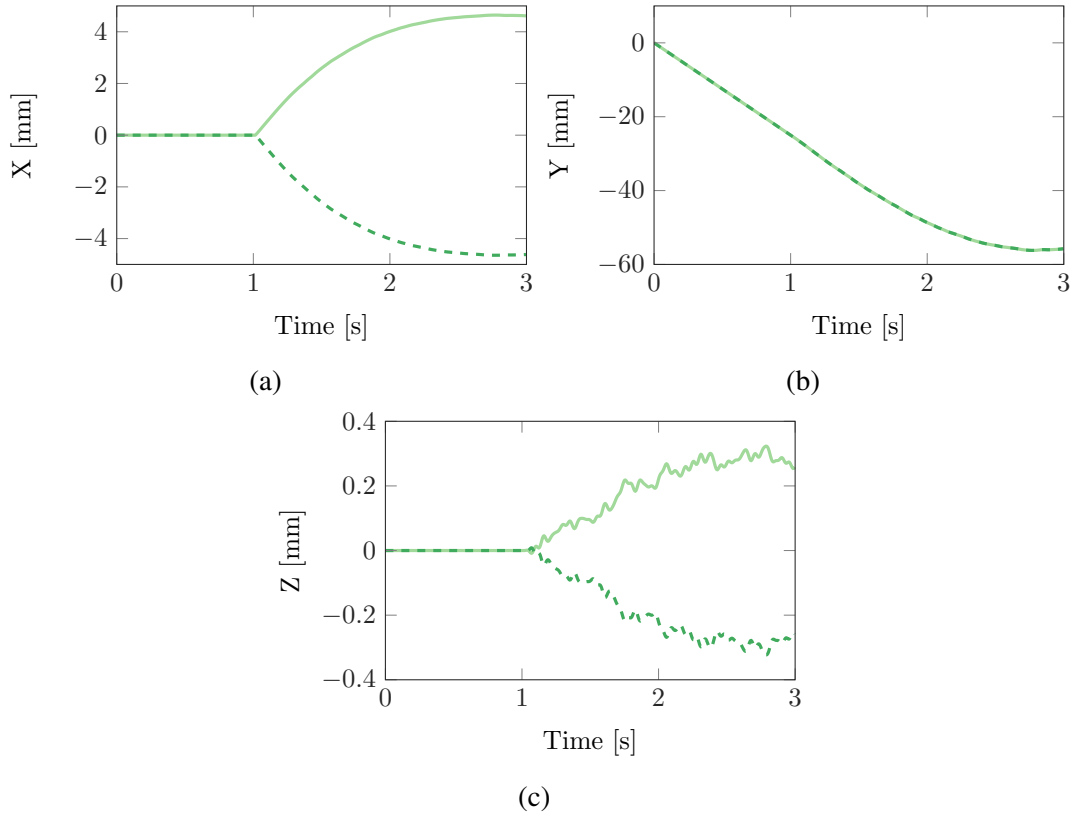


Fig. 4.5 Comparison of the displacement of the two nodes of the bottom face of the cube along X (a), Y (b) and Z (c). (—) Right Node, (---) Left Node.

4.2 Convergence Analysis

In this section, in order to validate the convergence of the results with the different numerical parameters, the Lagrange Multipliers method is used, without friction, without damping and with rigid rings. Analysis of the results concerning the method for the contact treatment, the friction, the damping and the flexibility of the rings are carried on in Sections 4.3 and 4.4.

4.2.1 Mesh of the rings

The first numerical parameter that requires to be set for the study is the mesh of the rings. It impacts the shape of the spline that interpolated the nodes of the mesh but not only. The size of the patches directly depends on the amount of nodes of the rings. It has its importance when it comes to the contact of the cube on those patches. The tool has to be able to deal with multiple nodes impacting the same patch for instance.

Four different meshes have been built, one very coarse (126 nodes), one coarse (264 nodes), one fine (1050 nodes) and one very fine (1872 nodes). All four are represented in Figure 4.6.

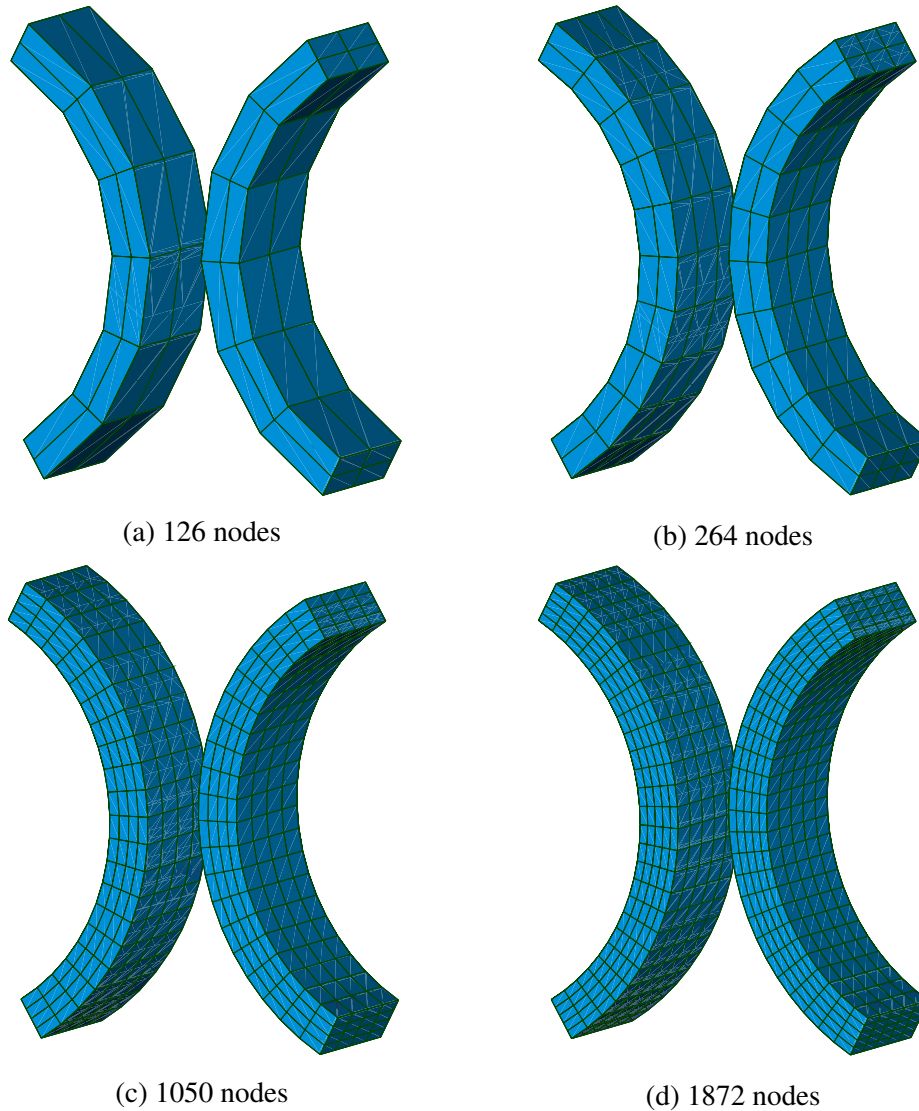


Fig. 4.6 Comparison of the different meshes for the rings.

The study has been realized not only on the mesh of the rings but on the mesh of the cube as well. However the impact on the algorithm of the latest is less significant. The discretization of the rings concerns the interpolation of the b-spline as well as the construction of the patches, two key variables when it comes to the simulation of the contact. Therefore, in this section solely the results for various meshes of the rings are presented while the mesh

of the cube remains identical.

The analysis of the results is shown on a single node of the cube positioned on its lower surface. It has been represented by a red dot in Figure 4.7. It has been chosen in particular since it is one of the first nodes to enter in contact with the rings and therefore is directly impacted by the mesh discretization.

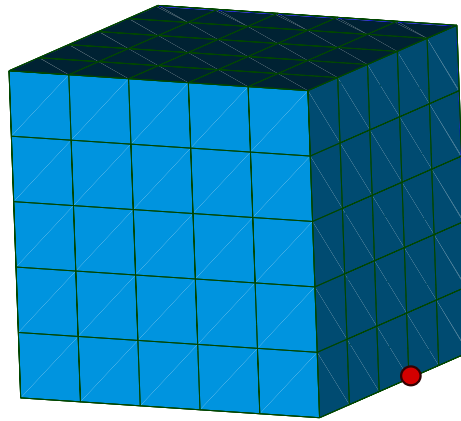


Fig. 4.7 Node of interest on the cube indicated by a red dot.

Figure 4.8(a) corresponds to the evolution of the distance between the rings and the node of the cube over time for the four meshes, Figure 4.8(b) to the evolution of the displacement of the node along the X axis, Figure 4.8(c) the displacement along the Y axis and Figure 4.8(d) along the Z axis, all for the four meshes.

Several conclusions can be directly drawn from those results. Firstly, from Figure 4.8(a) and Figure 4.8(c) one can observe that the contact duration and the progression between the rings depends on the mesh. The coarser the mesh the longer the contact and the deeper the progression. Secondly, from Figure 4.8(a) and 4.8(b), it is possible to see that the contact happens later on the very coarse mesh than on the others. These two observations are due to the construction of the b-spline. Coarse meshes means less point to interpolate for the construction of the b-spline. This leads to differences in the shapes for the b-spline. Even if there are only few millimetres of difference between the interpolations the effect can be non-negligible. The fact that the difference is doubled since it is repeated on the two rings is increasing the effect of a coarser mesh. Lastly, the differences between the fine and very fine meshes are so small that one can not differentiate the curves on the different graphics. Quantitatively, the maximum of the difference of the distances for the three coarser meshes

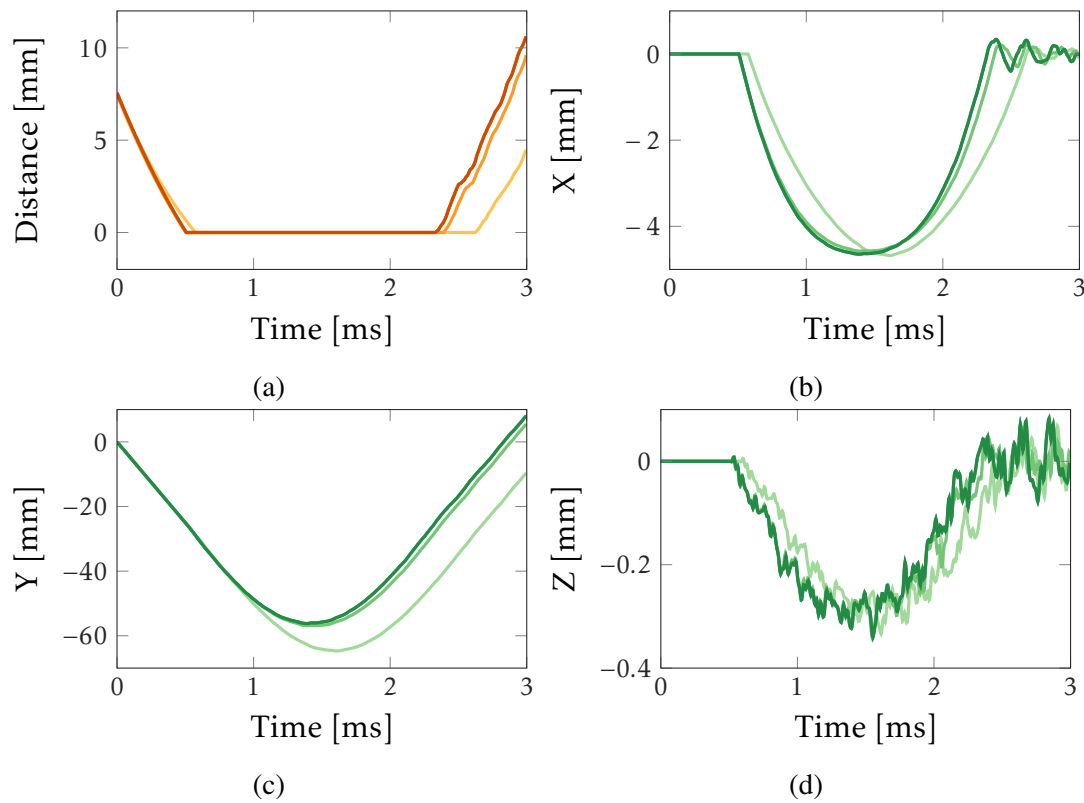


Fig. 4.8 Distance between the cube and the rings (a) and displacement of the cube along X (b), Y (c) and Z (d) for different meshes. (—, —) Very Coarse Mesh, (—, —) Coarse Mesh, (—, —) Fine Mesh, (—, —) Very Fine Mesh.

with respect the fine mesh are given in Table 4.2.

	Very Coarse - Very Fine	Coarse - Very Fine	Fine - Very Fine
Difference of distances	6.205	1.208	0.0016

Table 4.2 Difference of distances (in mm) between the different meshes and the finer mesh.

The graphics in Figure 4.8 as well as the results in Table 4.2 are, as anticipated, proving the convergence of the results with respect to the mesh. In the rest of this section, the mesh considered is the fine one, made of 1050 nodes (Figure4.6(c)).

4.2.2 Modal Reduction Parameter of the cube

The second studied parameter is the reduction parameter, i.e. the amount of constraint modes, retained in the Craig-Bampton method presented in Section 2.3.

Since this section only concerns infinitely rigid rings, the mass and stiffness of the rings are not playing a role in the simulations. Therefore, the cube is the only one reduced here. In Section 4.3, where the rings flexibility is considered, the reduction of their matrices will have an impact on the analysis.

The retained nodes are the ones on the face on the sides that enter in contact with the rings as well as the bottom face. The reduced cube can be seen in Figure 4.9.

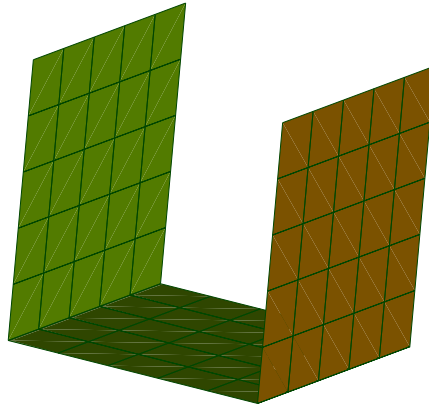


Fig. 4.9 Reduced model of the cube.

Since the focus is now on the cube rather than on the rings, the presentation of the results is arbitrary made for four nodes corresponding to the red dots that appear on Figure 4.10.

Those four nodes were picked in particular, since they belong to one of the two nodes that enter into contact with the rings. They are separated by 1 cm from each other. In addition, their positions with regard to one another helps to identify the impact of the contact of one node with the rings on the others.

The evolution of the distance between the rings and the four different nodes are presented in Figure 4.11.

From this figure, some interesting results can be provided. Firstly, concerning the comparison on the distances obtained with more or less constrained modes. The overall shape of the evolution of the distance over time is identical no matter how many constrained

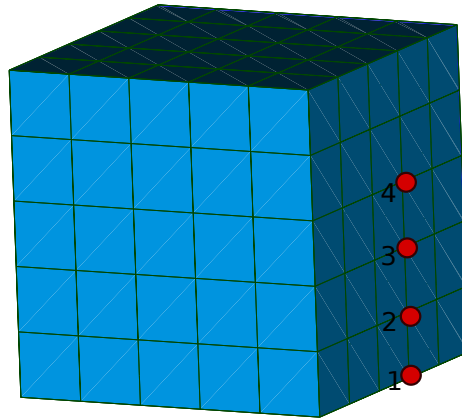


Fig. 4.10 Nodes of interest on the cube represented by the 4 red dots.

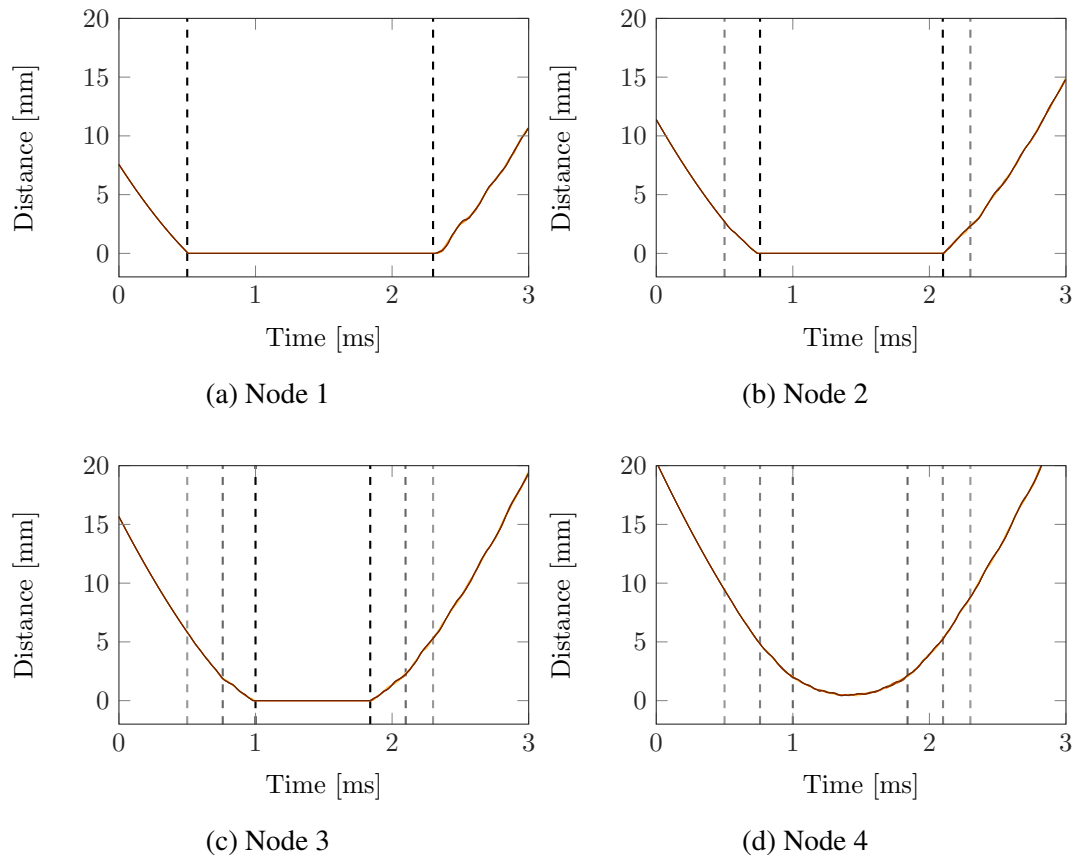


Fig. 4.11 Distance between the cube and the infinitely rigid rings for different reduction parameters. (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.

modes retained. However, a comparison of each result with the exact obtained solution considering all the nodes of the cube can be seen in Figure 4.12. Secondly, one can notice that the distance between the fourth node and the rings is strictly superior to zero, which means that this node does not enter in contact with the rings. Lastly, when one of the node enters in contact, the contact duration being delimited by the dashed lines in Figure 4.11 (corresponding to a duration of 1.8 ms for the contact of the first node, 1.3 ms for the second node and 0.9 ms for the third one), one can notice that some oscillations are appearing on the other nodes. This last effect is more visible in Figure 4.12. This simply corresponds to the response of the free nodes to the contact of the lower part of the cube.

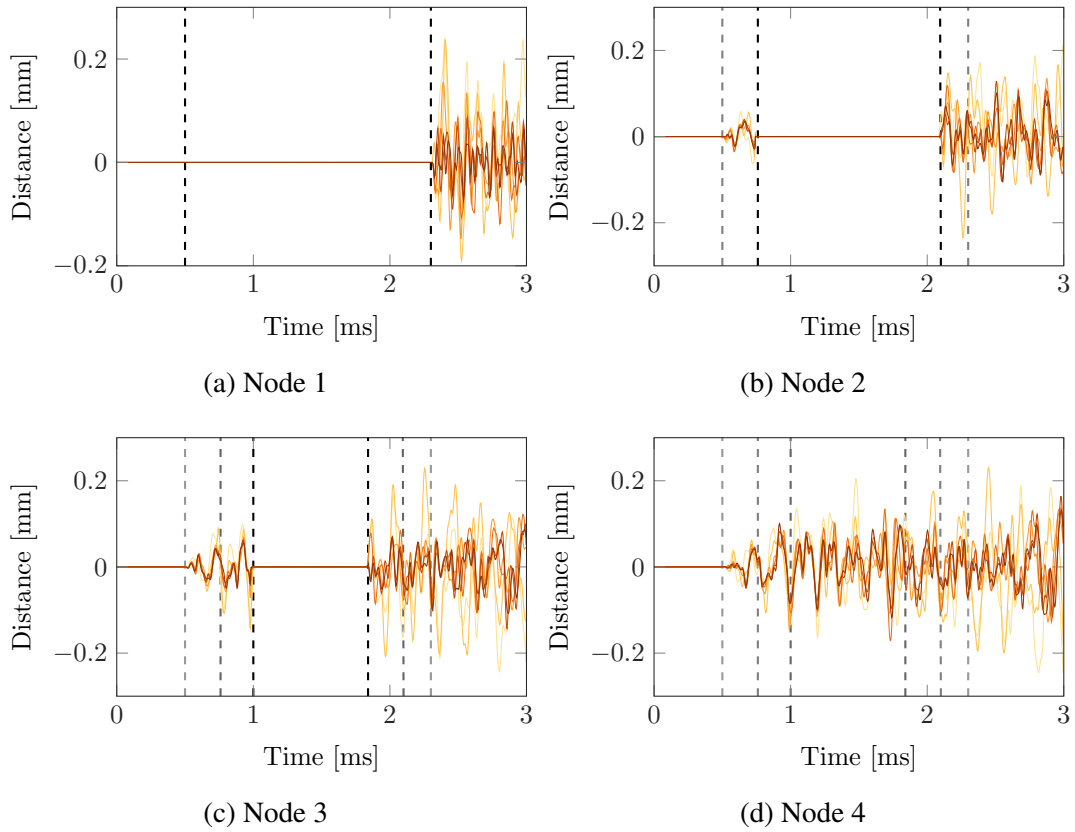


Fig. 4.12 Difference of distance between the reduced cube and the full one for different reduction parameters. (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes.

The differences between the curves now clearly appears. The more constrained-modes considered the closer the response to the full structure one is. Whenever there is a contact the distance is equal to zero no matter the reduction parameter, this is what can be seen in Figures 4.12(a), 4.12(b) and 4.12(c).

The vibrations due to the contact that were mentioned previously appear clearly in this figure. Between the first dashed lines in Figures 4.12(a) and 4.12(b), one can observe the presence of vibrations created by the contact of node 1 and its sudden stop due to the contact of the node studied.

In addition, to study the evolution of the distance regarding the reduction of the cube, the overall displacement of the same four nodes for the same reduction parameters is presented. Figure 4.13 corresponds to the displacement along X axis, Figure 4.14 along the Y axis and Figure 4.15 along the Z axis. Let's remind that the cube is moving along the Y axis.

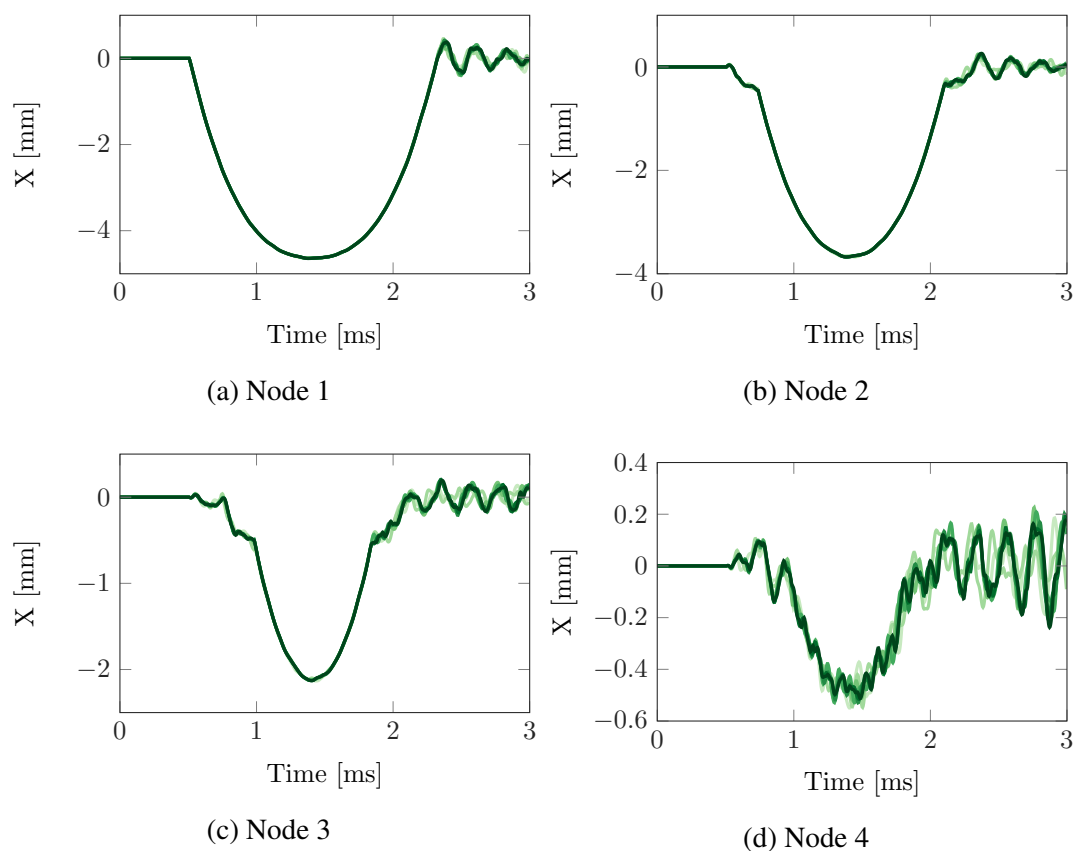


Fig. 4.13 Displacement along X of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.

These figures are proving the convergence of the solution with respect to the reduction. While the evolution of the displacement along Y (Figure 4.14) is highly similar depending on the reduction, this is not true for the displacement along X and Z (Figure 4.13 and Figure

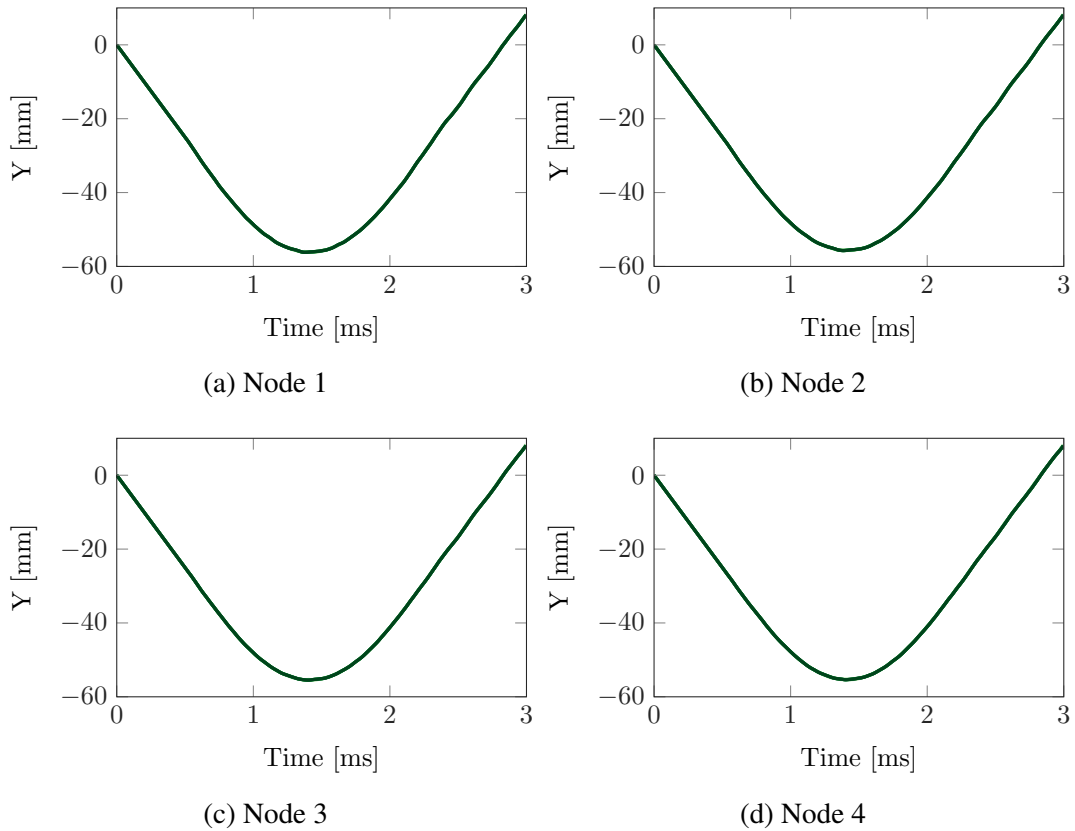


Fig. 4.14 Displacement along Y of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.

4.15). Indeed, one can differentiate the curves and observe the convergence of the results.

Concerning the dynamics of the system, when the cube enters in contact with the rigid rings, there is a compression due to the curvature of the rings this is what can be seen on Figure 4.13. This causes an expansion along the Z axis as Figure 4.15 shows it. One can notice that while the curves on Figure 4.13 are rather smooth, the ones on Figure 4.15 are not. This is due to the fact that along the X axis the cube displacement is limited by the rings geometry and the nodes are constrained to follow the curvature of the rings while along the Z axis their are not restricted in their motion. Therefore, as a consequence, the nodes can freely vibrate.

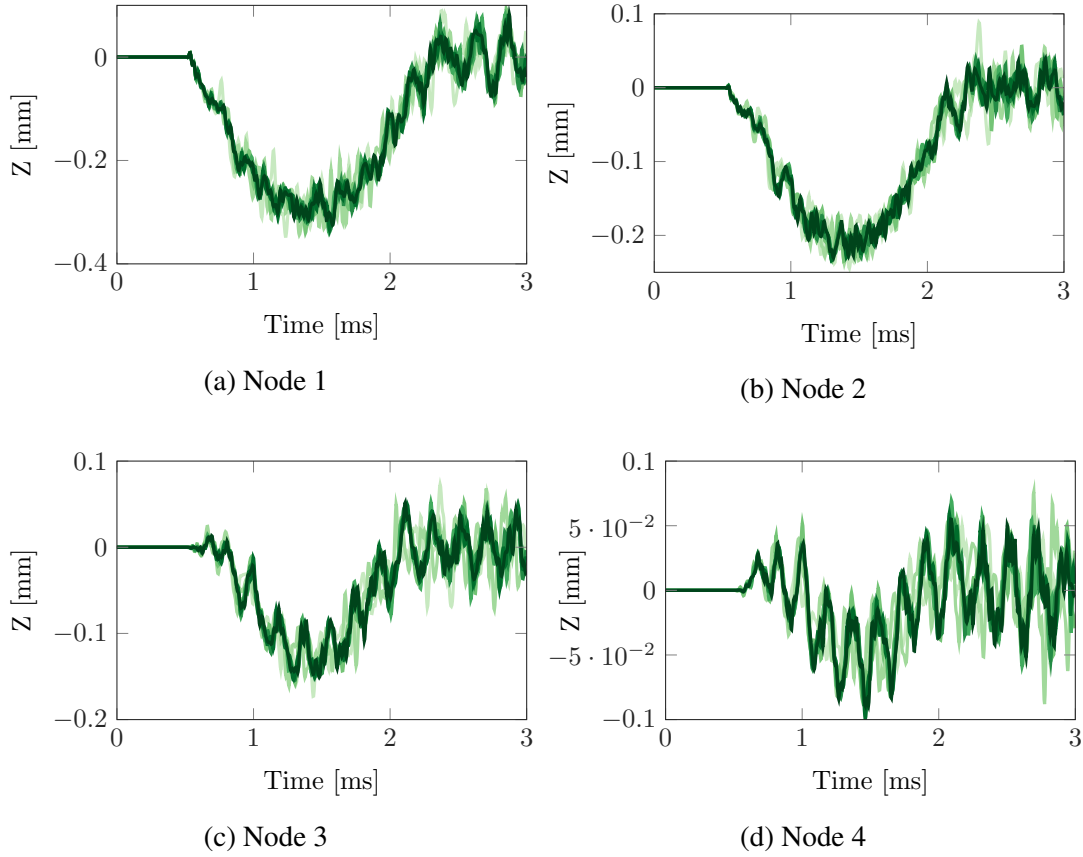


Fig. 4.15 Displacement along Z of the cube for different reduction parameters for node 1 (a), 2 (b), 3 (c) and 4 (d). (—) 0 constrained-modes, (—) 10 constrained-modes, (—) 50 constrained-modes, (—) 80 constrained-modes, (—) 100 constrained-modes, (—) 140 constrained-modes, (—) Full structure.

Along the Y axis the cube is simply falling between the rings until it bounces back. This corresponds to Figure 4.14.

A last aspect of the reduction has to be taken into account, it concerns the computation time. All the computation for this study were realized on a Intel®Core™i7-6700K CPU @ 4.00GHz x4 with 15.5 GiB of RAM.

For the evaluation of the computation time two scenario are considered. The first one is when none of the nodes are in contact and the second one when the first row of nodes are in contact (12 nodes). The comparison of the computation time corresponds to Figure4.16(a) and Figure4.16(b).

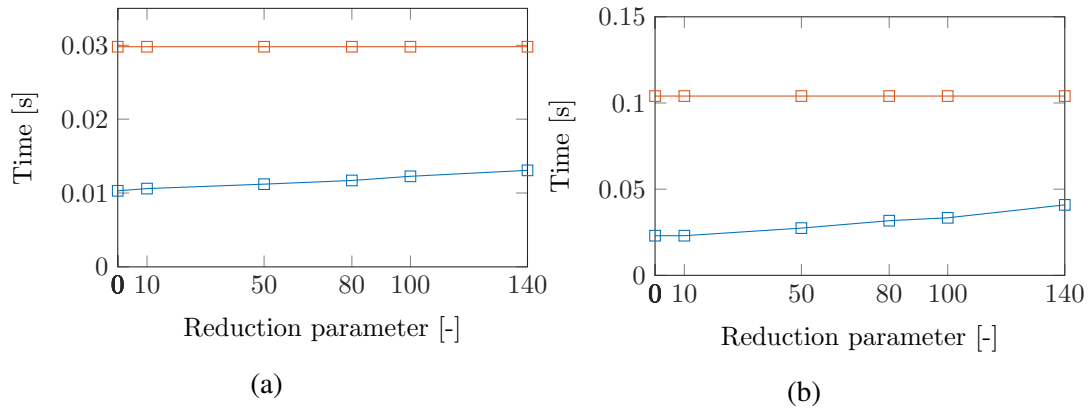


Fig. 4.16 Comparison of the average computation time of one iteration with respect to the reduction parameter without contact (a) and during the contact (b).

There is a non negligible difference between the non-reduced and reduced structure which by itself justify the reduction that is applied to the system studied. This is due not only to the size of the mass and stiffness matrices that are different, but mostly because the distance is evaluated for every node of the cube. Furthermore, looking at the time axis in Figure 4.16(b), one can notice that during contact the computation time of one time step is about three times more important. This is due to the contact treatment and the correction calculation that is adding time each iteration. Finally, let's mention that this time increase is depending on the amount of considered nodes in contact. The more the nodes in contact the longer the computation is.

4.2.3 Time step

The next analysis is related to the time step. This is a very sensitive parameter when it comes to contact behaviour. For instance if the time step is too large, the contact is only detected after significant penetrations of one body into the other one. This can lead to non-physical behaviour and eventually algorithm resolution issues.

Looking at the temporal evolution of the distance between the cube and the rigid rings one can draw Figure 4.17.

It clearly appears that the distance evolution for a time step equal to $5 \cdot 10^{-6}$ s is not following the same path as the others. Indeed, significant penetrations are detected between two iteration times. This large time step also induces the cube to leave the contact later than it

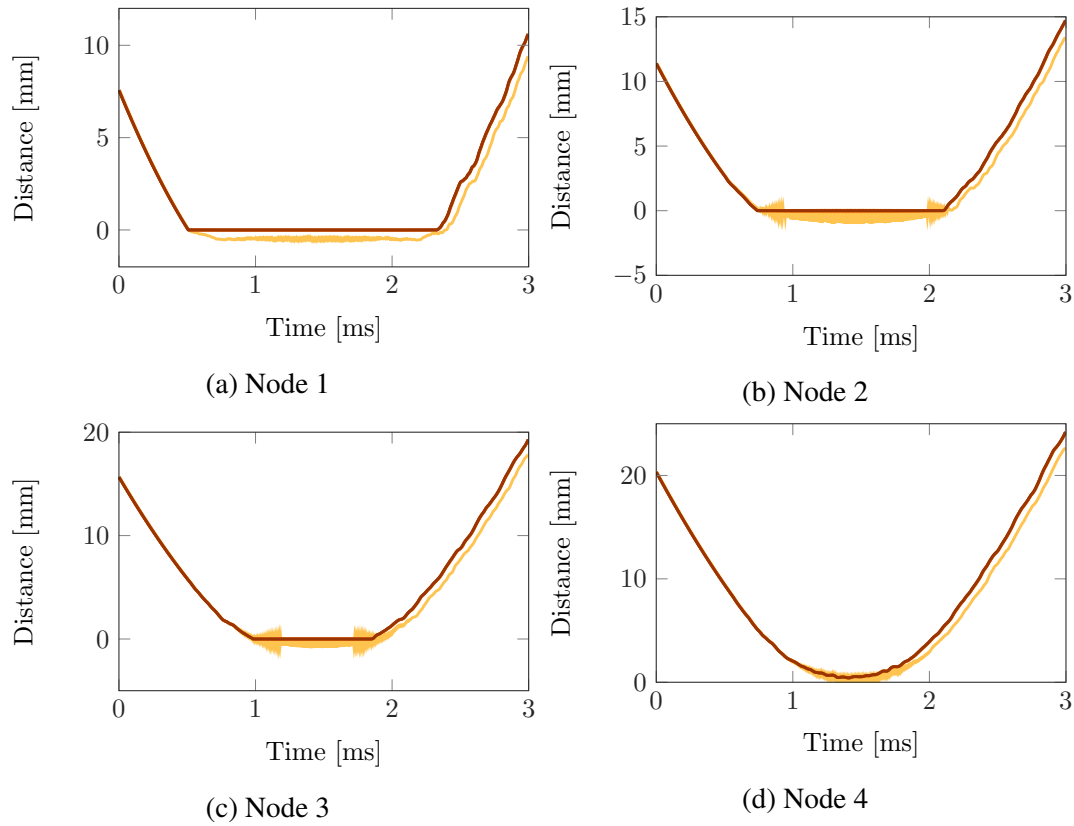


Fig. 4.17 Distance between the cube and the infinitely rigid rings for different time steps. (—) $h = 5 \cdot 10^{-6}$ s, (—) $h = 1 \cdot 10^{-6}$ s, (—) $h = 5 \cdot 10^{-7}$ s, (—) $h = 1 \cdot 10^{-7}$ s, (—) $h = 5 \cdot 10^{-8}$ s.

actually does. The information on when exactly the contact is over is delayed. This explains the fact that the curve corresponding to the larger time step has the overall same shape as the others after the contact even though it is time translated. Let's mention that if one wants to realize an analysis with a time step greater than $5 \cdot 10^{-6}$ s the numerical instabilities would be too important and the computation would stop. On the other hand smaller time step could have been studied but even though the time step does not increase the computation time of one iteration it increases the amount of iteration. Thus, the total computation time can be significantly impacted by the choice of the time step.

Taking a closer look at the contact period between 0.95 ms and 1.9 ms, one can obtain Figure 4.18.

This zoom on the different sub-figures allows to highlight the convergence of the results with respect to the time step. It clearly appears, and in particular in Figure 4.18(b), that the solution is converged for a time step of $1 \cdot 10^{-7}$ s.

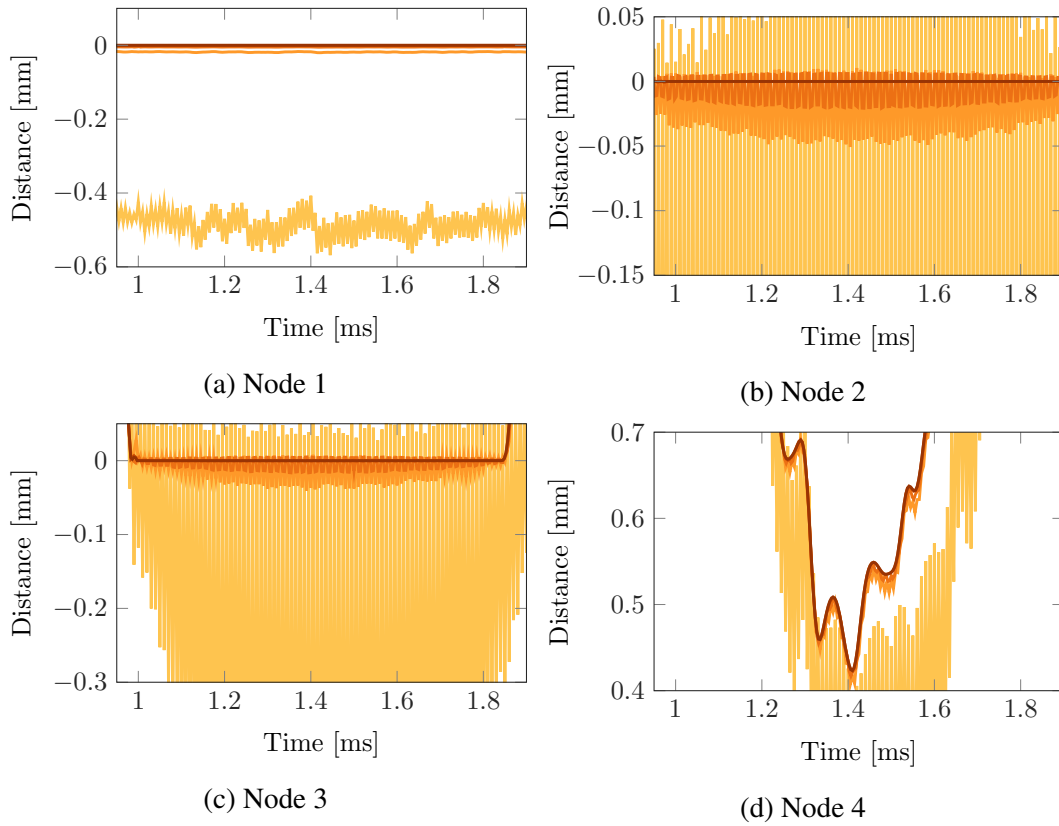


Fig. 4.18 Zoom on the distance between the cube and the infinitely rigid rings for different time step. (—) $h = 5 \cdot 10^{-6}$ s, (—) $h = 1 \cdot 10^{-6}$ s, (—) $h = 5 \cdot 10^{-7}$ s, (—) $h = 1 \cdot 10^{-7}$ s, (—) $h = 5 \cdot 10^{-8}$ s.

4.2.4 Newton-Raphson coefficient

The last numerical parameter that can be discussed is the stop criterion of the Newton-Raphson (N-R) algorithm presented in section 2.5.1.

This parameter impacts the precision at which one decides that the closest point of the nodes of the cube on the surface has not moved of a certain factor between two iterations of the N-R algorithm. However, compared to the mesh, the time step and the reduction parameter, the stop criterion parameter is less significant.

Indeed, the analysis of the evolution of the distance for the four nodes is exactly identical for parameters equal to 10^{-2} m up to 10^{-5} m, which means that the algorithm detects the same closest points for those specific parameters. This is the case at least for the four nodes studied, some others may have their closest points slightly moved. In order to obtain a new closest point a parameter equals to 10^{-6} m had to be set. This means that this new point is

only a few micrometers away from the previous one.

Since this algorithm starting point at each time step is the position of the closest point at the previous time step, the cost of this extra precision is nearly zero. Thanks to the small time step between two iterations, the point found at a previous time is really close to the one at the next time step. Therefore, the cost of this extra precision is limited to the very first iteration. For instance, the average computation time for the first iteration can be found in Table 4.3.

N-R Coefficient [m]	Time [s]
10^{-2}	0.0988
10^{-3}	0.0991
10^{-4}	0.0994
10^{-5}	0.10095
10^{-6}	0.1023

Table 4.3 Computation time for the first iteration of the tool with different N-R coefficient.

This last table shows that the computation time cost for the Newton-Raphson parameter is negligible compared to what can be obtained with the reduction method.

4.3 Analysis of the results

Now that the influence of the value of the numerical parameters in order to obtain a converged solution have been set, the different physical parameters can be presented.

4.3.1 Flexible Rings

The first parameter that can be change concerns the rigidity of the rings. So far, the rings were supposed infinitely rigid which means that they by definition could not be deformed in anyway.

If now one considers the exact same scenario has previously, with a cube of steel falling at the same velocity of 50 m/s without friction nor damping. But this time with deformable rings. Let's remind that the mechanical properties of the rings are identical to the ones of the cube and that they can be found in Table 4.1. The dynamic of the system is totally different and more representative of an actual physical case. However, the flexibility of

the impacted structure means that in case of contact the nodes of this structure are subjected to movement. The cost of that is that the b-spline interpolating those nodes has to be re-computed at each time step which can become costly depending on the amount of nodes.

Looking at the evolution of the distance over time as well as the displacement along the three axis of the same node as the one presented in Figure 4.7 for flexible rings one can obtain Figure 4.19.

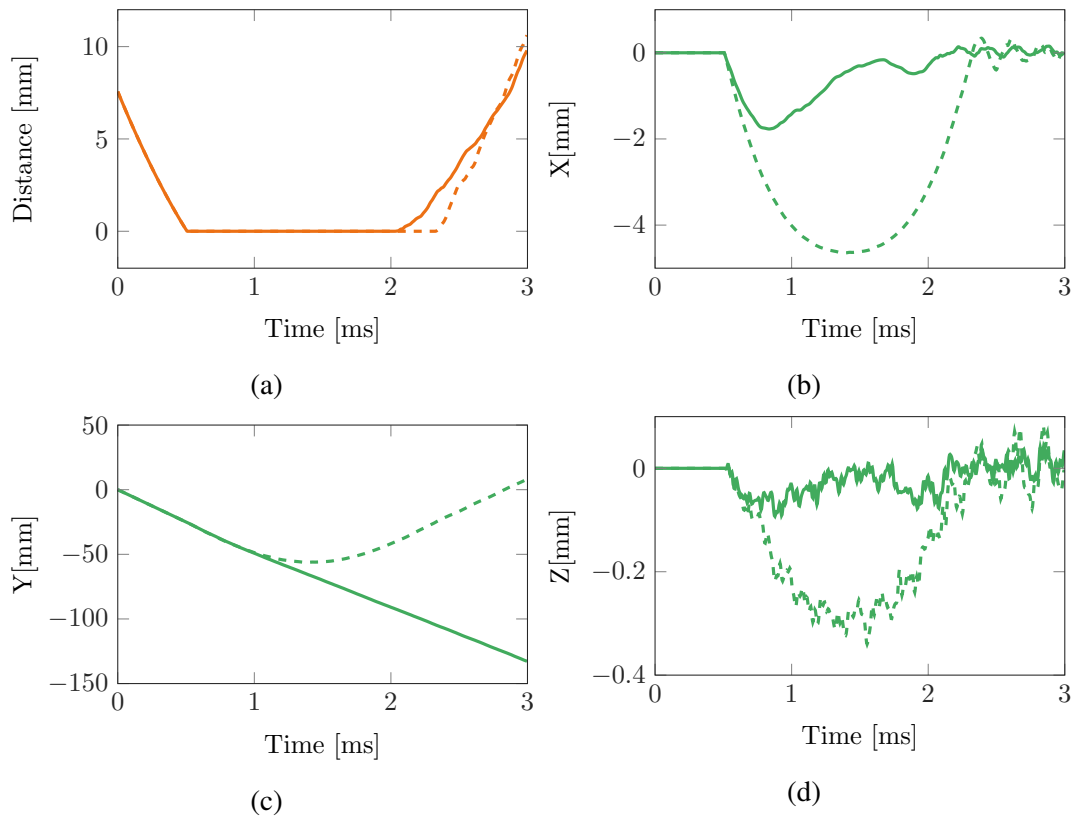


Fig. 4.19 Distance between the cube and the flexible rings (a). Displacement of the node of the cube along X (b), Y (c) and Z (d). (---) Rigid, (—) Flexible, (---) Rigid, (—) Flexible.

The main information about the dynamic of the system comes for Figure 4.19(c) where one can see that that displacement along Y is strictly downward. This means that the cube does not bounce on the rings like it was the case for rigid rings but deforms them to get through.

In addition to that, the displacement along the X axis is significantly lower for flexible rings than it was for the rigid ones. The same goes for the displacement along Z. While

the first is compensated by the displacement of the rings, the reduced displacement along Z comes from less constraint contact.

A last parameter that can be compared for the rigid and the flexible rings is the contact effort that is computed in order to correct a penetration. This is what Figure 4.20 corresponds to.

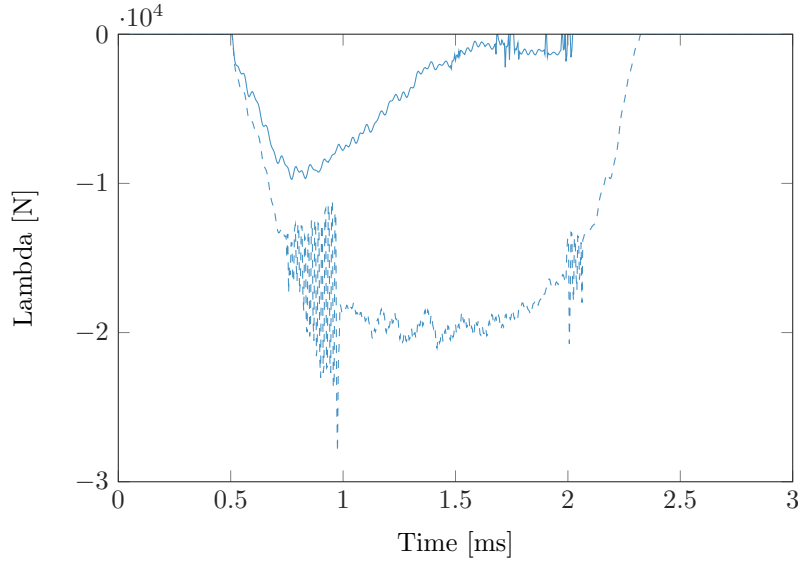


Fig. 4.20 Comparison of the contact efforts, λ , for rigid rings (---) and flexible rings (—).

As can be seen, the contact forces with flexible rings are lower than for the rigid case. This anticipated result comes from the fact that the two rings are following the cube thus softening the impact.

Exactly like it was the case for the cube, the flexible rings can be reduced using the Craig-Bampton method. In addition to that the time step and the mesh can affect the response. However, the analysis concerning the time step or the mesh as already been previously conducted in Section 4.2 concerning the rigid rings, and the conclusions have been drawn. The reduction of the rings is still to be analysed in order to set the appropriate reduction parameter.

Due to the contact with the cube, the ring gap is suddenly increasing, as can be seen in Figure 4.21(a). The reduction of displacement between the rigid and the flexible for the cube node seen in Figure 4.19(b) is compensated by this displacement.

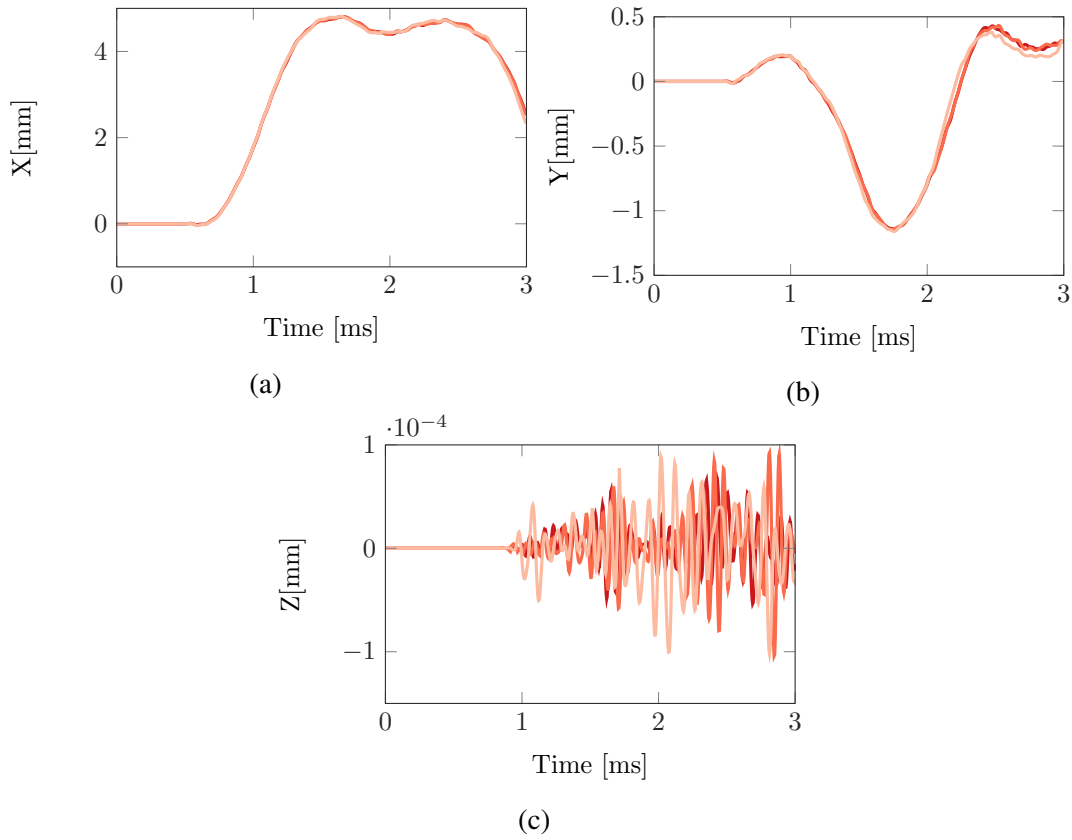


Fig. 4.21 Displacement of one node of the flexible rings along X (a), Y (b) and Z (c) for different reduction parameters. (—) 0 constrained modes (—) 100 constrained modes, (—) 200 constrained modes.

Along the Y axis, since the impact is slightly above the node studied, the displacement is first upward before the cube communicates its downward movement. Since the rings are clamped at their extremities, the displacement is limited in the space. Moreover, since the motion of the cube is purely along the Y axis, the movement along the Z axis can be neglected.

4.3.2 Damping

At this stage the topic of damping have been left on the side. All the simulation have been done with a damping matrix set to 0. However, as described in the theoretical chapter, the damping matrix plays an important role in the Newmark integration scheme and therefore the prediction of the displacement.

Including the no damping case, three types of damping are studied. The two others are respectively the proportional damping and the modal damping. The importance that the user

of the developed tool desires to give to the damping is depending on the user's choice. In this section the impact of damping in general is discussed. The equations related to the two methods are briefly introduced.

Proportional Damping

The proportional damping depends on the two first eigen-frequencies f_1 and f_2 of the damped structure. The values ε_1 and ε_2 , corresponding to a percentage, depend on the user decision.

$$b = \left(2\varepsilon_2 - 2\varepsilon_1 \cdot \left(\frac{f_1}{f_2} \right) \right) / \left(\left(\frac{1}{f_2} \right) - \left(\frac{f_2}{f_1^2} \right) \right) \quad (4.1)$$

$$a = \left(2\varepsilon_1 - \frac{b}{f_1} \right) / f_1 \quad (4.2)$$

$$\mathbf{D} = a \cdot \mathbf{K} + b \cdot \mathbf{M} \quad (4.3)$$

Modal Damping

This method is based on the computation of the eigen-values ϕ , and the eigen-vectors \mathbf{v} . The parameter ξ impacts the intensity of the damping and depends on the user choice.

$$\mathbf{D}_m = 2 \cdot \xi \cdot \text{diag}(\sqrt{\phi}) \quad (4.4)$$

The damping matrix is given by:

$$\mathbf{D} = \text{inv}(\mathbf{v}^T) \times \mathbf{D}_m \times \text{inv}(\mathbf{v}) \quad (4.5)$$

Figure 4.22 corresponds to the comparison of the displacement of the bottom node of the cube previously studied with damping taking into account or not. The results displayed here are obtained with the proportional damping assumption with the condition that the error made on the 2 first modes is less than 1%.

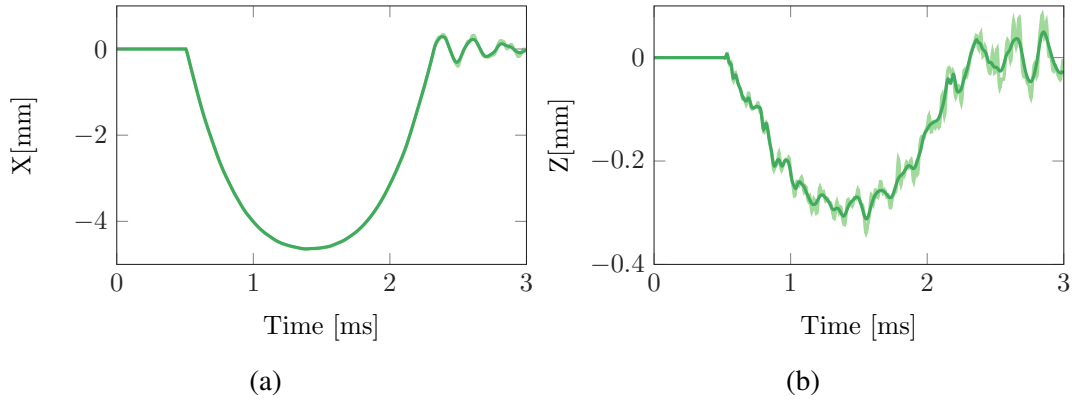


Fig. 4.22 Displacement of the node of the cube along X (a) and Z (b). (—) Damped, (---) No Damping.

The effect of the damping is to reduce the vibrations. Those vibrations are mainly present on the displacement along X and Z . One can observe that the curves with damping added are smoother than the one without. This result, yet elementary, is important since it proves that the software is correctly simulate the damping of a system.

The same analysis has been conducted with flexible rings. The results are available in Figure 4.23.

Globally the same conclusion can be drawn and the consideration of the damping for both rigid and flexible structure is validated for the numerical tool. Its effect is well apparent in Figure 4.23(e), representing the displacement along Z of the rings. The addition of the damping to the system clearly shows that the displacement along that direction is equivalent to zero.

4.3.3 Friction coefficient

So far, the friction between the rings and the cube has been neglected. If the friction coefficient μ is different of zero, a tangential component is added to the contact treatment (see section 2.5.2 equation 2.31).

Figure 4.24 represents the evolution of the distance as well as the displacement of the cube along the three axis for different values of μ for rigid rings.

Three different behaviours can be described from Figure 4.24. The first one corresponds to the transition from $\mu = 0$ to $\mu = 0.1$. The introduction of a small friction coefficient does not radically changed the dynamic of the system. The cube is still falling and bouncing after

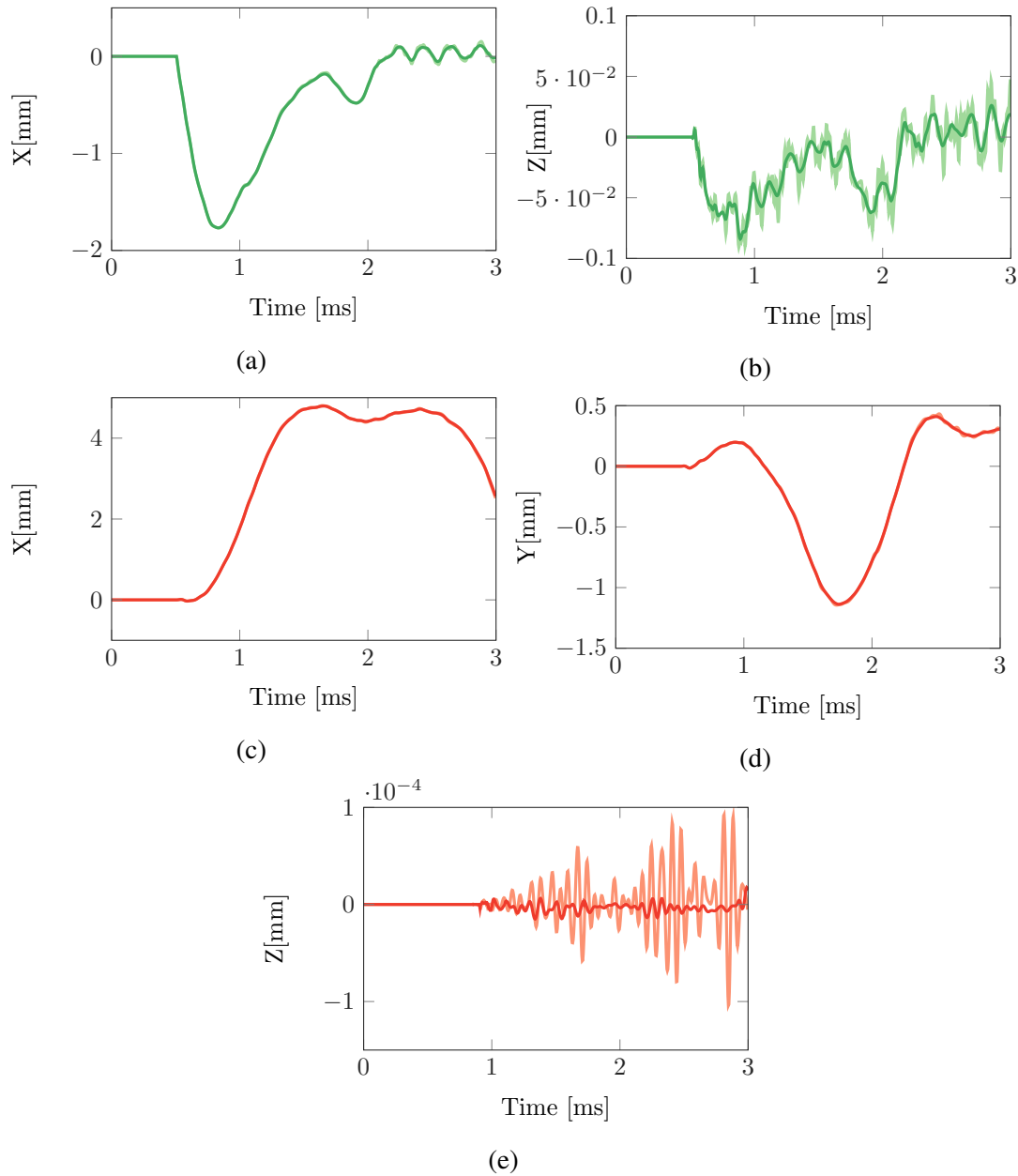


Fig. 4.23 Displacement of the node of the cube along X (a) and Z (b) Displacement of the node of the rings along X (c), Y (d), Z (e). (—, —) Damped, (—, —) No Damp.

having been deformed by the rings. However the contact last longer, the deformation is less important and the progression through the rings is less deep.

For a medium value of the friction coefficient, $\mu = 0.3$, the cube stays in permanent contact with the rings, or in other words is stuck between the rings. The cube, when entering in contact with the rings, is first slowed down but yet pursues its progression between them.

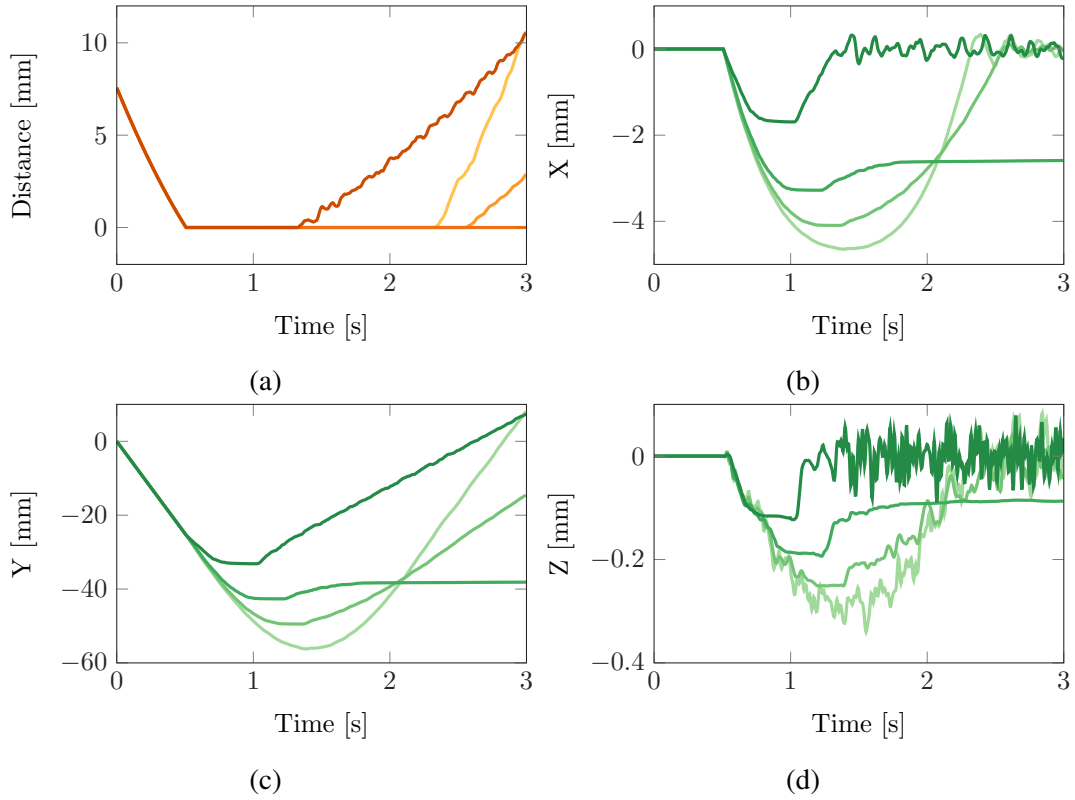


Fig. 4.24 Distance between the cube and the rings for different friction coefficient (a). Displacement of the node of the cube along X (b), Y (c) and Z (d) for different friction coefficient. (—, —) $\mu = 0$, (—, —) $\mu = 0.1$, (—, —) $\mu = 0.3$, (—, —) $\mu = 0.8$.

At some point, the friction has dissipated so much energy that the cube stops in a deformed state. But the geometry of the rings is such that the cube is pushed upward in order to regain its original shape. This is corresponding the small increase of the displacement before it's stabilized (Figures 4.24(b), 4.24(c), 4.24(d)).

Lastly, for a higher value of the friction coefficient, $\mu = 0.8$, when the lower part of the cube hits the rings it is almost immediately stopped. The rest of the cube is still moving inducing a sudden compression of the cube. Following that compression there is an equivalent expansion making the cube bounce. This is what can be seen in Figure 4.24(b). The duration of the contact is shorter than the case without friction and so is the after bounce velocity.

The same type of analysis can be applied to the case of flexible rings. This corresponds to Figure 4.25.

The introduction of a low friction coefficient has the same effect than for rigid rings, the global dynamic is not changed. The cube is losing its velocity during its contact with the two

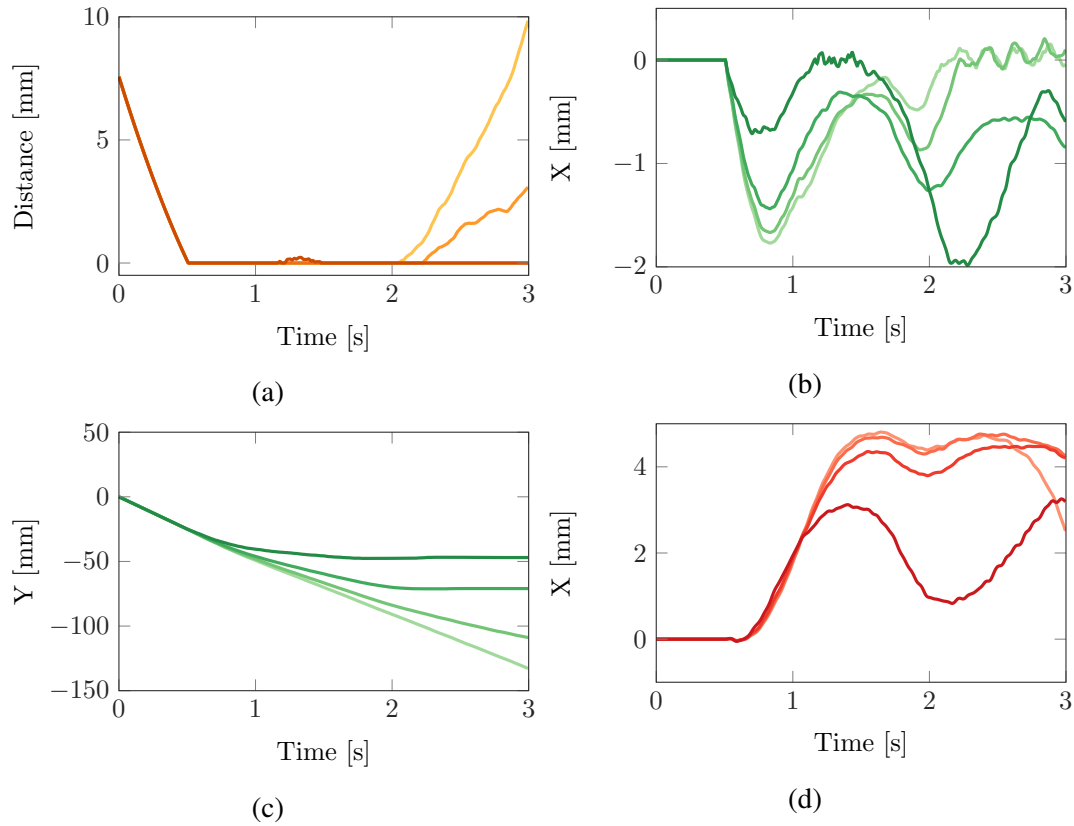


Fig. 4.25 Distance between the cube and the rings for different friction coefficient (a). Displacement of the node of the cube along X (b), Y (c) and displacement of the node of the ring along X for different friction coefficient (d). (—, —, —) $\mu = 0$, (—, —, —) $\mu = 0.1$, (—, —, —) $\mu = 0.3$, (—, —, —) $\mu = 0.8$.

rings but has still enough energy to progress through them.

Like it was the case for rigid rings, with a medium value of μ (0.3), the cube is slowed enough to stop between the rings. Let's notice that this stop happens deeper with flexible rings than rigid rings (Figures 4.24(c) and 4.25(c)).

Last but not least, when $\mu = 0.8$, the cube is also suddenly stopped but instead of compressing as much as for the rigid rings, a part of that compression is transmitted to the rings. Once the compression of the cube is turning into an expansion, the rings are still moving away due to that initial contact to the point that the contact is interrupted for a brief instant (Figure 4.25(a)). After that the cube stays stuck between the rings it starts to oscillate with them.

4.4 Comparison between Penalty and Lagrange Multipliers methods

The last aspect of the developed tool that must be presented and validated concerns the method to compute the correction of the eventual penetration tied to the contact. As mentioned in Section 2.5.2, two methods have been implemented: The Lagrange Multipliers and the Penalty method.

Figure 4.26 corresponds to the distance, the effort and the displacement of the node displayed in Figure 4.7 for the two methods. Values for the penalty coefficient, from 10^6 to 10^{10} , are presented.

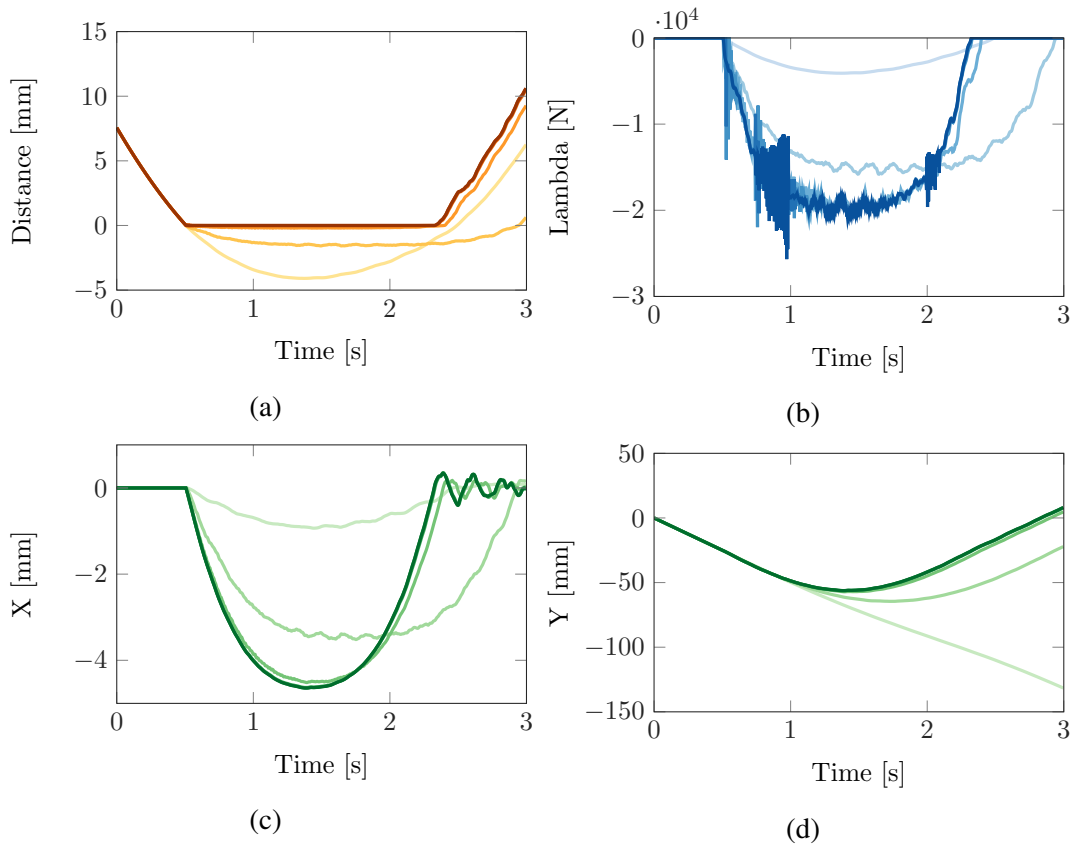


Fig. 4.26 Distance between the cube and the rigid rings for the two contact methods (a). Contact forces for the two contact methods with rigid rings (b). Displacement of the node of the cube along X (c), Y (d) for the two contact methods with rigid rings. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Penalty $k_p = 10^9$, (—, —, —) Penalty $k_p = 10^{10}$, (—, —, —) Lagrange Multipliers.

As can be seen in Figure 4.26(a), the values of the distance between the cube and the rings are negative for some values of k_p . This is due to the fact that the Penalty method, contrary to the Lagrange Multiplier, allows the penetration and therefore negative distances. This penetration depends on the value of k_p , the smaller the k_p the more important the penetration. However, if the k_p is too small, the algorithm becomes unstable and eventually crashes. On the other hand, if the k_p is too large, the applied correction is too responsive. This can lead to instabilities as well.

Looking at Figure 4.26(d), in particular for the case of $k_p = 10^6$ one may think that the system is behaving like for the flexible rings despite them being rigid. In fact, the contact forces are so small, that the cube is falling through the rings almost like they were not there in the first place. This behaviour is obviously not physically correct.

This misrepresentation of the physics introduced by the Penalty method can be corrected by an appropriate choice of coefficient. As Figure 4.26 shows, for k_p greater than 10^8 , the curves have converged toward the Lagrange Multipliers results.

The Penalty method gets its advantage over the Lagrange Multiplier method from its simplicity. Indeed, while for the Penalty method the contact forces, λ , are simply resulting from a scalar multiplication of the distances of the node in contact by the k_p , the Lagrange Multiplier method requires matrix manipulation such as multiplication and inversion (Section 2.5.2).

Now looking at the impact of those two methods on the flexible rings case, Figure 4.27 can be drawn.

Overall same type of conclusions can be obtained. However, the impact a small penalty coefficient is less pronounced due to the motion of the rings that soften the response of the cube. Comparing Figures 4.27(b) and 4.26(b), one can notice that the contact forces are about two times smaller for the flexible case than for the rigid one. The impact of smaller coefficient is therefore reduced.

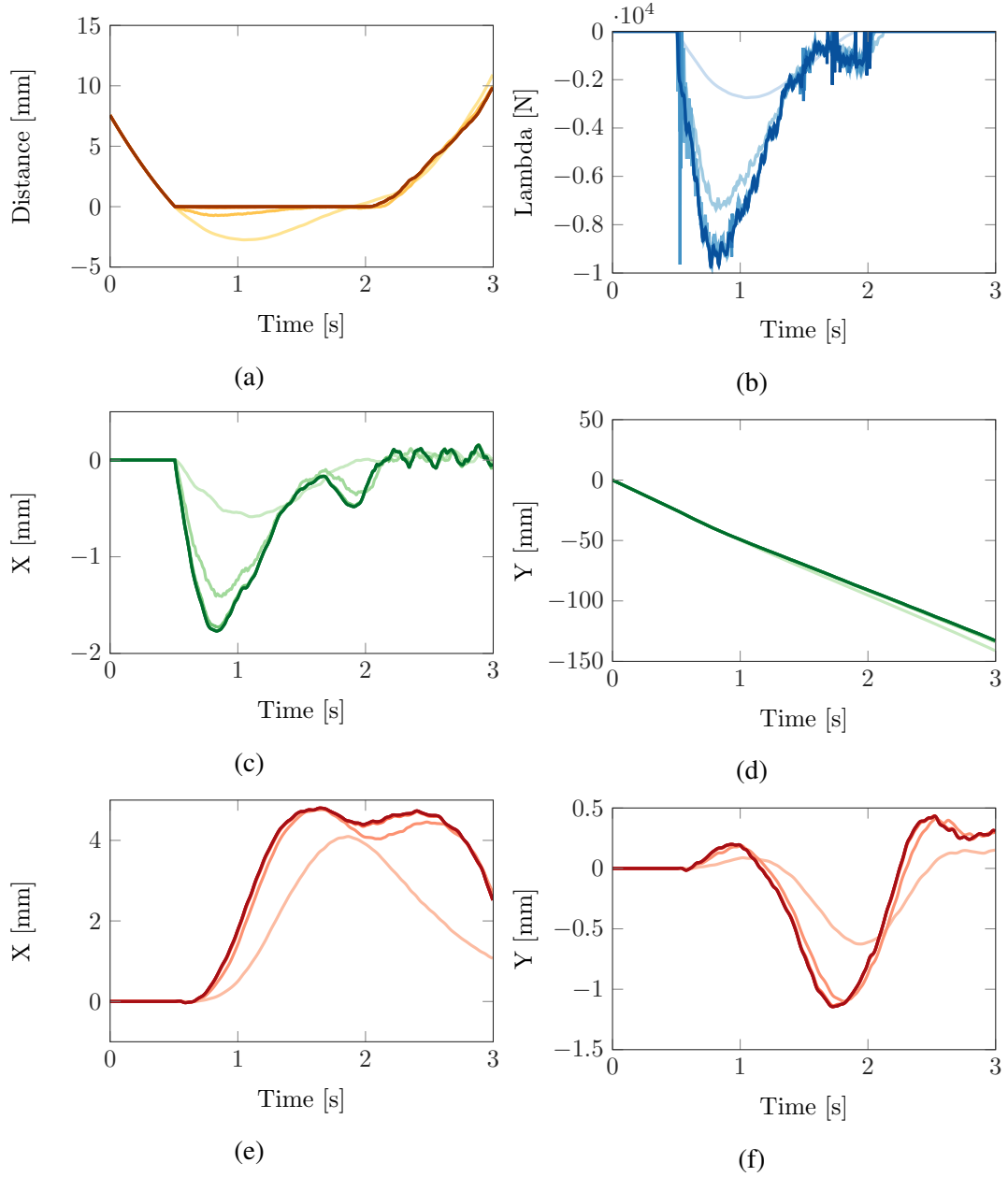


Fig. 4.27 Distance between the cube and the flexible rings for the two contact methods (a). Contact forces for the two contact methods with flexible rings (b). Displacement of the node of the cube along X (c), Y (d) for the two contact methods with flexible rings. Displacement of the node of the ring along X (e), Y (f) for the two contact methods with flexible rings. (—, —, —, —) Penalty $k_p = 10^6$, (—, —, —, —) Penalty $k_p = 10^7$, (—, —, —, —) Penalty $k_p = 10^8$, (—, —, —, —) Penalty $k_p = 10^9$, (—, —, —, —) Penalty $k_p = 10^{10}$, (—, —, —, —) Lagrange Multipliers.

Conclusions

This chapter demonstrated that the developed tool during this work is able to realistically represent a wide range of situation starting from a simple cube falling between two rings.

The dependence with the numerical parameters has been presented and allowed to highlight the convergence of the results regarding the time step or the reduction method. In addition, the tool proved that even on coarse meshes the contact is properly managed even though the global geometry of the rings depends on the amount of nodes.

The introduction of the flexibility, the damping as well as the friction lead to interesting scenario such as permanent contact state with the cube stuck between the rings or even deformation of the rings to let the cube go through them. This aspect being validated, the adaptation and the representation including the physics of the contact between many bodies can be envisaged.

Finally, the two methods to treat the contact, the Penalty method and the Lagrange Multipliers, have been validated in the case of rigid or flexible rings. The pro's and the con's of each have also been highlighted.

Now that the various numerical and physical parameters have been validated throughout a systematic analysis, the tool that has been developed can be employed to simulate a more complex situation corresponding to the contact of a turbine blade in rotation with a casing.

Chapter 5

Interaction of contact between a compressor blade and its casing

Introduction

An aircraft engine is a very complex device, it can roughly be decomposed in 5 main sections: the low-pressure compressor, the high-pressure compressor, the combustion chamber, the high-pressure turbine and the low-pressure turbine. Engines can be more complex but every modern aircraft engines include those different parts. Numerous blades are composing turbines and compressors. A scheme of an aircraft engine showing the different parts is represented in Figure 5.1.

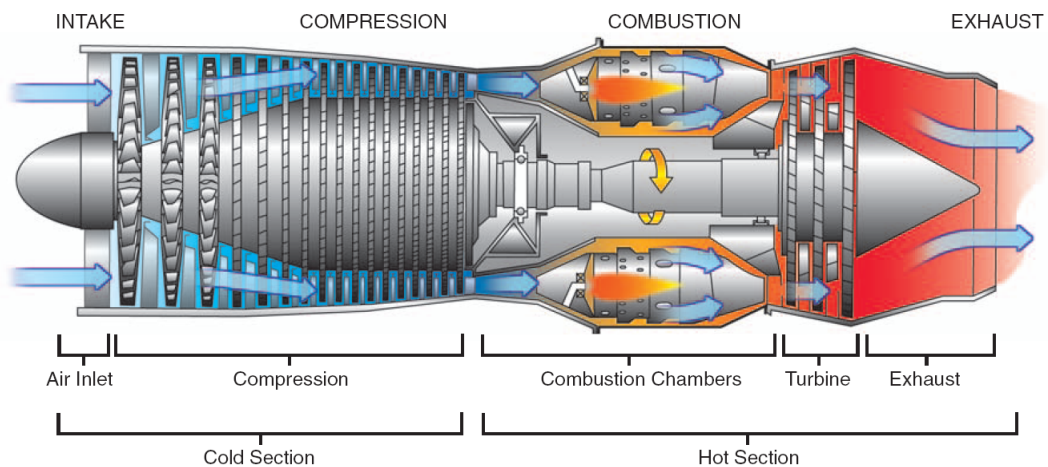


Fig. 5.1 Basic components of a gas turbine engine [21].

Under high rotation speed of the engine, the air is compressed, heated and then expanded before it exits the engine. The blades are working in a hostile environment, under high pressure and high temperature. A failure in the engine can cost numerous lives. Furthermore, the requirements in terms of environmental impact of the aircraft engines are constantly increasing. For all those reasons, the turbine and compressor blades are continuously studied and tested in order to improve the performances of the engines and insure the safety of the aircraft passengers.

Due to the high temperature, deformations of the casing surrounding the engine or the blades can occur. This can eventually leads to contact interaction between the blades and the casing. This particular topic is addressed in this chapter.

The first section includes the description of the geometry, the boundary conditions of the casing and compressor blade studied as well as its reduction.

The second part of this chapter concerns, in a first time, the analysis of the contact between the blade in rotation and the rigid casing. The Lagrange Multipliers and the Penalty method are compared first regarding the penalty coefficient then regarding the rotation speed. In a second time, the simulations are reproduced considering a flexible casing.

5.1 Description of the simulation case

The blade studied in this chapter is a compressor blade, with the corresponding finite element model represented in Figure 5.3. The blade is rotating inside its casing represented in Figure 5.2. This casing has four of its nodes clamped represented by the four yellow dots corresponding to the fixations of the engine. The inner part of the casing is the part modelled with the b-spline surface inside the numerical tool. It corresponds to a total of 420 nodes. The blade is clamped as well on its lower part, the nodes concerned by this constraint are represented by yellow dots in Figure 5.3. The ones represented in red are corresponding to the nodes retained for the reduction of the model.

The finite element model represented in Figure 5.3 are made of 22 898 nodes in total, corresponding to 68 694 degrees of freedom. The reduction of the system allows to greatly simplify the computation since only 8 nodes are retained.

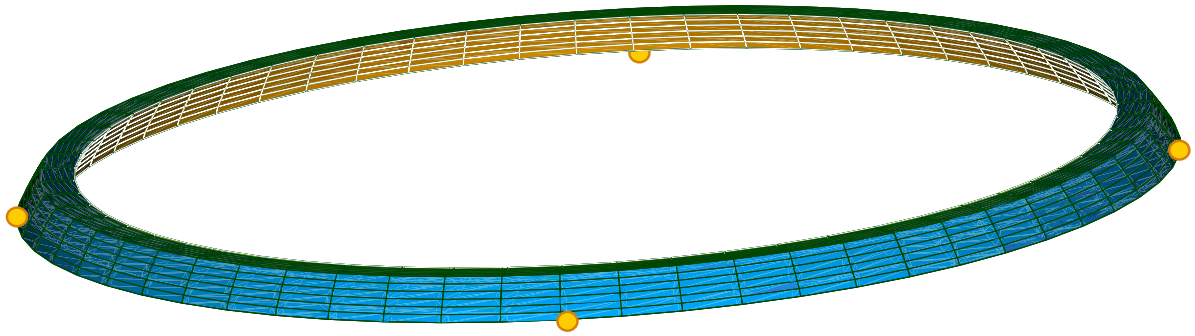


Fig. 5.2 Finite element model of the compressor blade casing with the clamped nodes represented by yellow dots.

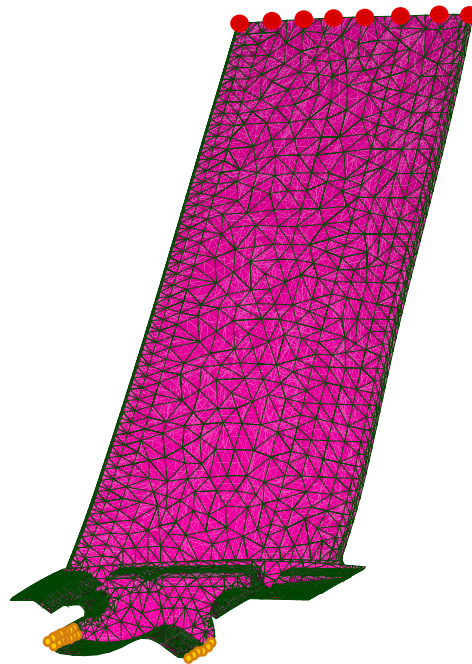


Fig. 5.3 Finite element model of the turbine blade with the retained nodes represented by red dots and clamped nodes represented by yellow dots.

In order to determine the amount of constrained modes that must be considered for the Craig-Bampton method to properly represent the whole structure, the eigen frequencies obtained for different reduction coefficients with the one from the full structure can be compared. The frequencies associated to the ten first eigen-modes of the full compressor blade and the reduced one are compared in Figure 5.4. The associated relative errors can be found in Table 5.1.

As the relative errors are showing it, the simple reduction of the blade, without constrained modes, provides satisfying results for the first frequency only. However, as soon as

the frequency increases, the error does as well. Increasing the amount of constrained modes up to 5, the results are significantly improved. The error on the frequency is ten times smaller than it was with 0 constrained mode. Since the error on the first mode has been reduced significantly with only taking 5 constrained modes, one can seek with how many constrained modes this error drops down to zero, it happens for exactly 10 constrained modes, which, seen the other percentages of error, has been the value considered for the future analysis carried out in this chapter. The resulting model is a 34 d.o.f. system while without the reduction the system had 68 694 d.o.f.. It is more than 2000 times less than the full structure.

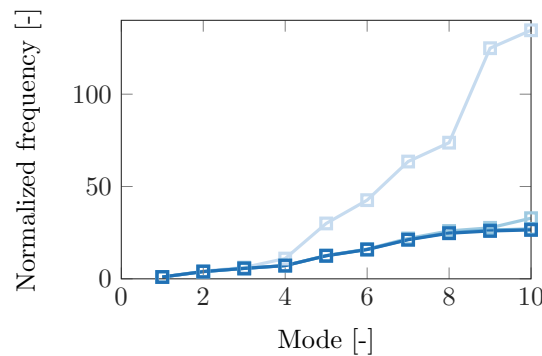


Fig. 5.4 Convergence of the eigen-mode of the turbine blade with respect to the amount of constrained mode considered in the reduction. (—) 0 constrained-modes, (—) 5 constrained-modes, (—) 10 constrained-modes, (—) Full structure.

	C-B 0	C-B 5	C-B 10
Relative error on mode 1	0.568 %	0.055%	0%
Relative error on mode 2	4.95 %	0.17%	0.0156%
Relative error on mode 3	9.22 %	0.259%	0.076%
Relative error on mode 4	53.24 %	0.5369%	0.094%
Relative error on mode 5	140.20 %	0.9275%	0.224%
Relative error on mode 6	168.79 %	1.406%	0.562%
Relative error on mode 7	200.05 %	3.33%	0.66%
Relative error on mode 8	198.70%	5.05%	1.182%
Relative error on mode 9	381.87 %	6.55%	2.047%
Relative error on mode 10	408.53 %	24.05%	1.939%

Table 5.1 Relative errors on each mode between the reduction for different reduction parameters and the full structure.

The graphical representation of the blade (reduced at its retained nodes) turning, at a velocity Ω , inside the casing (also reduced) in the numerical tool is corresponding to Figure

5.5.

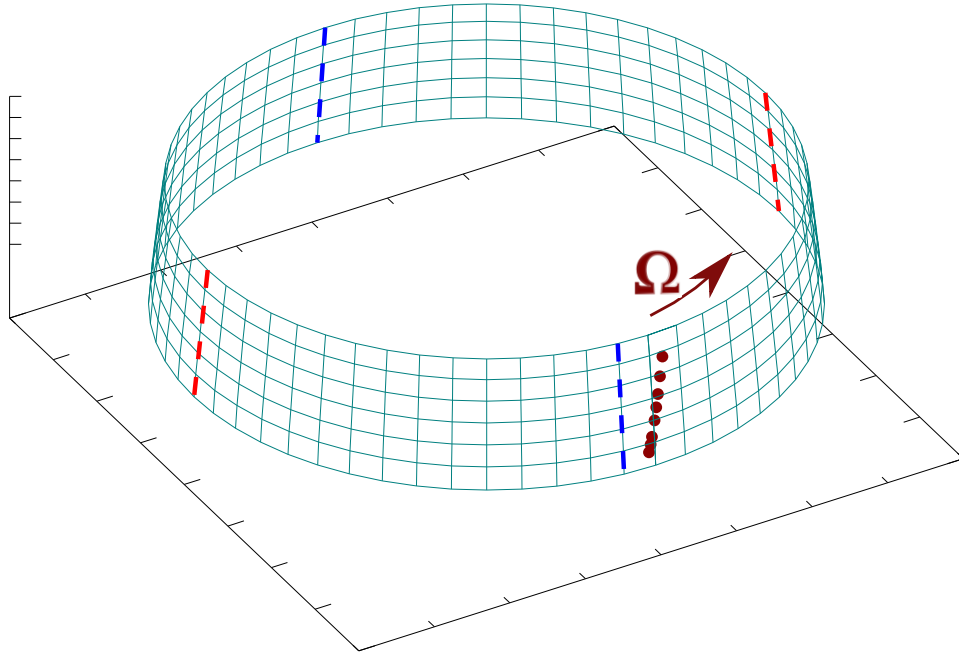


Fig. 5.5 Representation of the retained nodes of the compressor blade inside the casing in the developed tool.

Let's mention that the casing is not perfectly circular otherwise no contact could have been possible. There is a small gap between the nodes of the blade and the casing. This gap maximum at the positions of the blue dashed lines and minimum on the red dashed lines in Figure 5.5. At its maximum, the gap prevents the contact between the blade and the casing. At its minimum, it is the opposite, the gap is such that the blade is entering in contact with the casing. The casing is not a perfect cylinder but has a conical shape. The gap at the leading edge of the blade larger than the one at the trailing edge. This way the blade has, in theory, two contact phases with the casing.

5.2 Analysis of the results

5.2.1 Rigid casing

In this section, the displacements of the nodes of the casing are neglected and the casing is considered as totally rigid. In a first time, the Lagrange Multipliers and the Penalty

method are compared for the same rotation speed. Using the results presented in the previous chapter, the time step have been set to 10^{-7} s and the stop parameter of the Newton-Raphson algorithm to 10^{-6} . In addition to that, since the friction coefficient has to be as low as possible in practice, no friction have been considered in this chapter. Furthermore, small modal damping have been introduced in the system in order to stabilized the response of the blade. First looking at the leading edge of the compressor blade, one can obtain the results represented in Figure 5.6.

Globally, the same type of observations that were made in the previous chapter can be repeated in this case as well. First regarding the evolution of the distance over time, one can see that with the lower values of the penalty coefficient, k_p , there is a significant penetration of the blade in the casing. In total 40 contact phases can be counted from the different figures of Figure 5.6. This corresponds to 2 contacts per rotation which is due to the geometry of the casing. Furthermore, the transition of the response toward a steady response, for which the amplitude of the variation remains constant, clearly appears in Figure 5.6(d). The steady state is reached after approximately 3 rotations, or 6 contact phases. Even though the amplitude of the displacements and the amplitude of the evolution of the distance are remaining constant after those 3 rotations, the amplitude of the contact forces represented in Figure 5.6(b) does not seem to converge toward a constant value. This is only true for the Lagrange Multipliers method. Indeed, the Penalty method relies on the multiplication of the prediction of the displacement by a constant. This prediction being steady the contact forces are therefore too. Regarding the Lagrange Multipliers method, the vibration of the blade leads to rapid transitions between contact/no-contact phases for each node, this impacts the construction of the contact matrices presented in Section 2.7.

Taking a closer look at all the figures presented in Figure 5.6, considering one complete, one can draw Figure 5.7. This figure clearly displays the different curves. For instance, the constant amplitude of the contact forces obtained thanks to the Penalty method are well apparent in Figure 5.7(b). A very interesting fact can directly be derived from that last observation. Despite the fact that the contact forces obtained with the Lagrange Multipliers and the Penalty method with a $k_p = 10^8$ are so different from one another, the response are matching almost perfectly (Figures 5.7(a), 5.7(c), 5.7(d) and 5.7(e)). The difference between the displacements obtained via the two methods is less than $1 \mu\text{m}$ at its maximum. In order to validate that this value of k_p is the most appropriate for a detailed comparison of the two methods, one can take a look at the results obtained at the trailing edge.

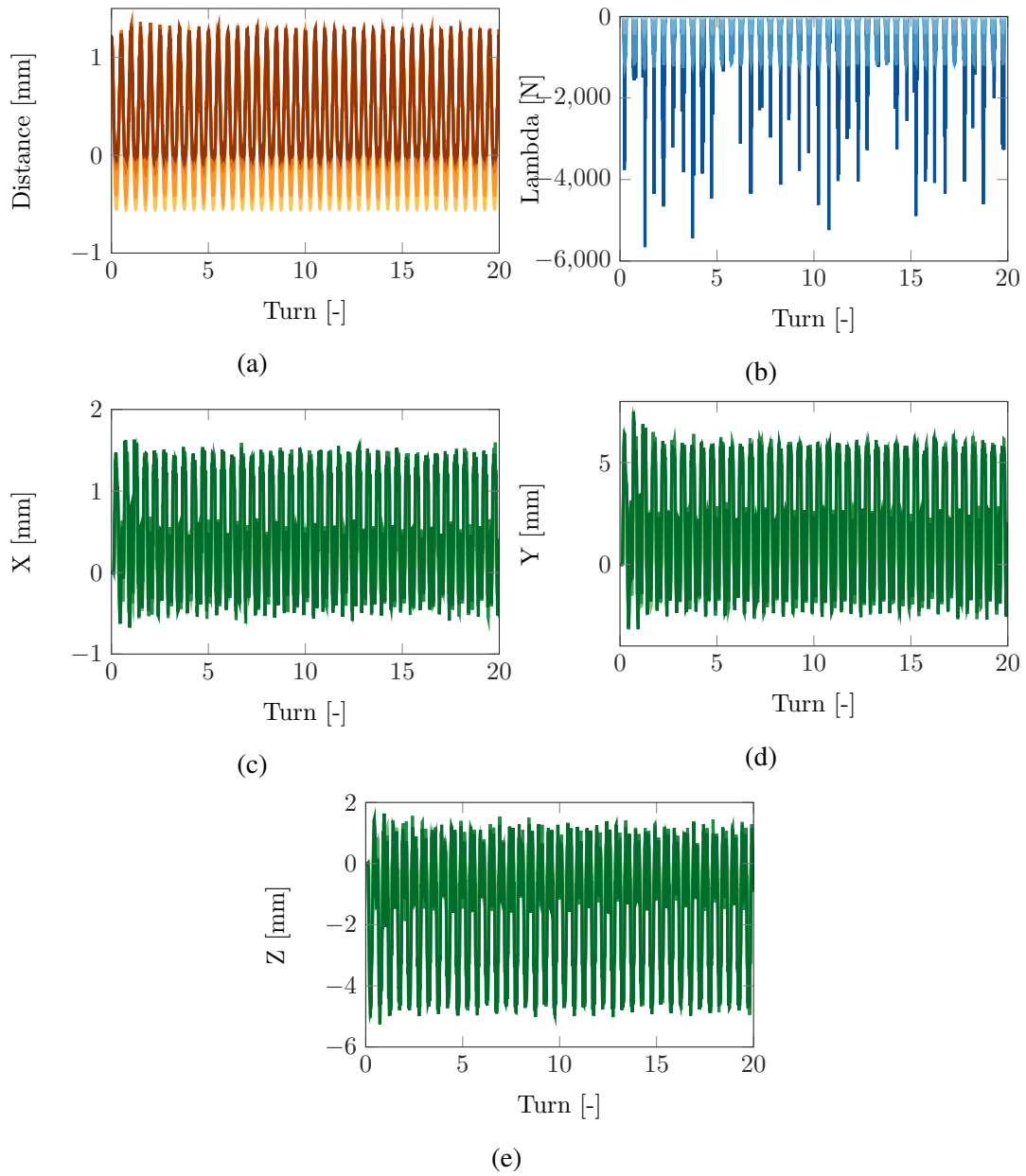


Fig. 5.6 Distance between the leading edge of the compressor blade and the rigid casing for different contact algorithms (a). Contact forces at the leading edge of the compressor blade for different contact algorithms (b). Displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

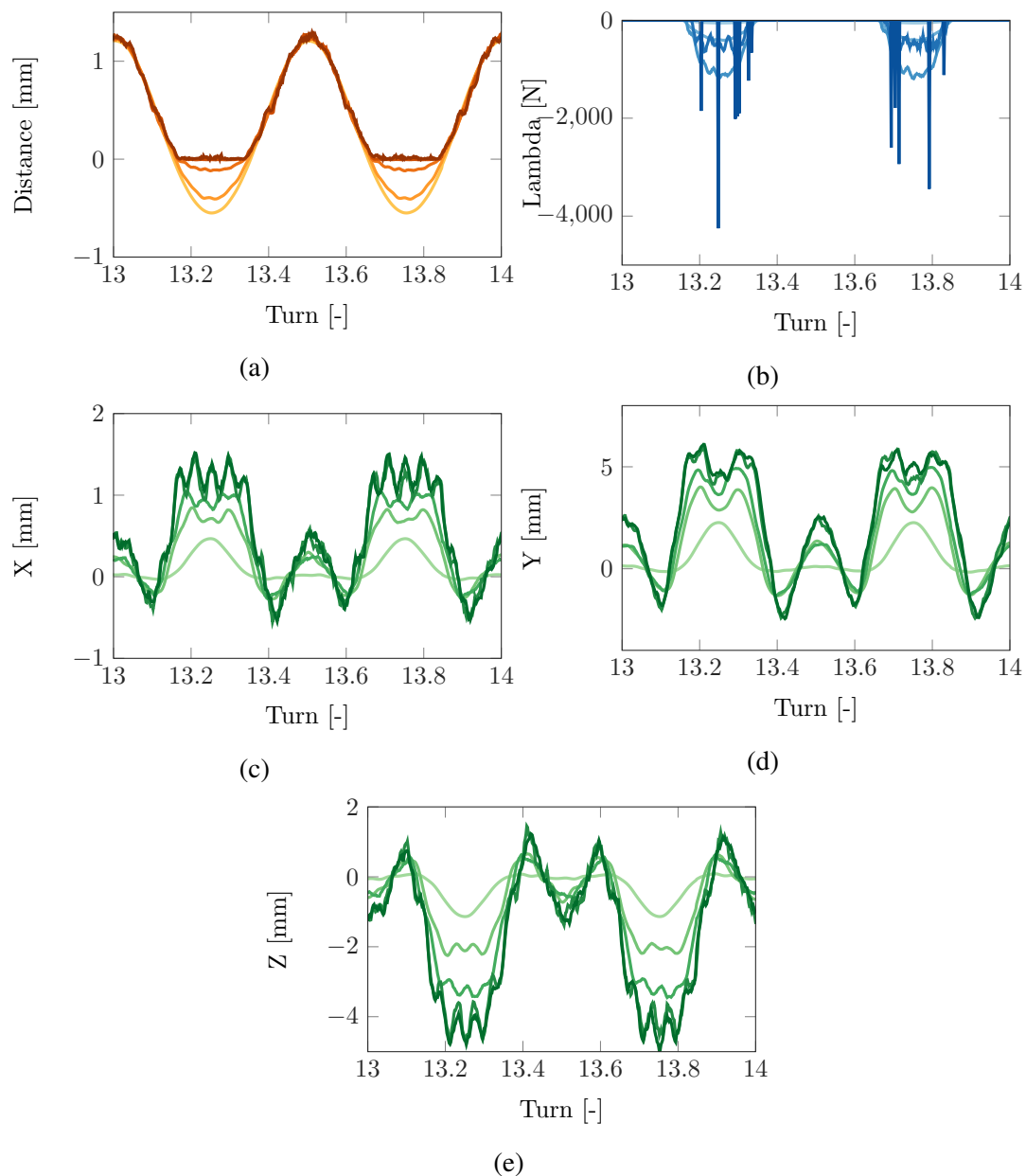


Fig. 5.7 Zoom on the distance between the leading edge of the compressor blade and the rigid casing for different contact algorithms (a). Zoom on the contact forces at the leading edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

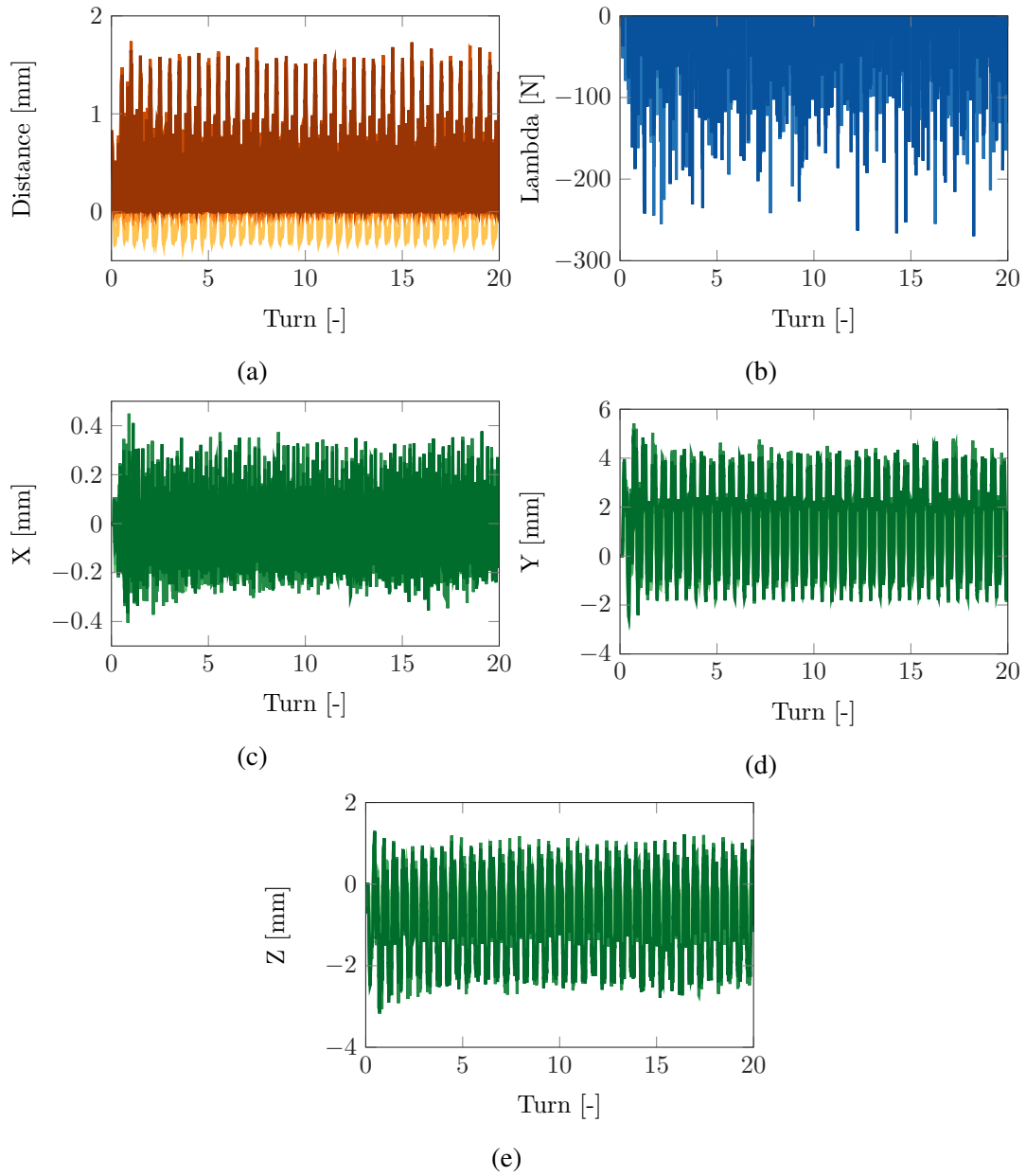


Fig. 5.8 Distance between the trailing edge of the compressor blade and the rigid casing for different contact algorithms (a). Contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

Figure 5.8 corresponds to the evolution of the distance, the contact forces and the displacement of the trailing edge node of the blade. An interesting result that directly appears on those figures is that a total of 120 contact interactions can be counted during the 20 full

rotations. That represents a total of 6 contacts per rotation in opposition to the 2 on the leading edge.

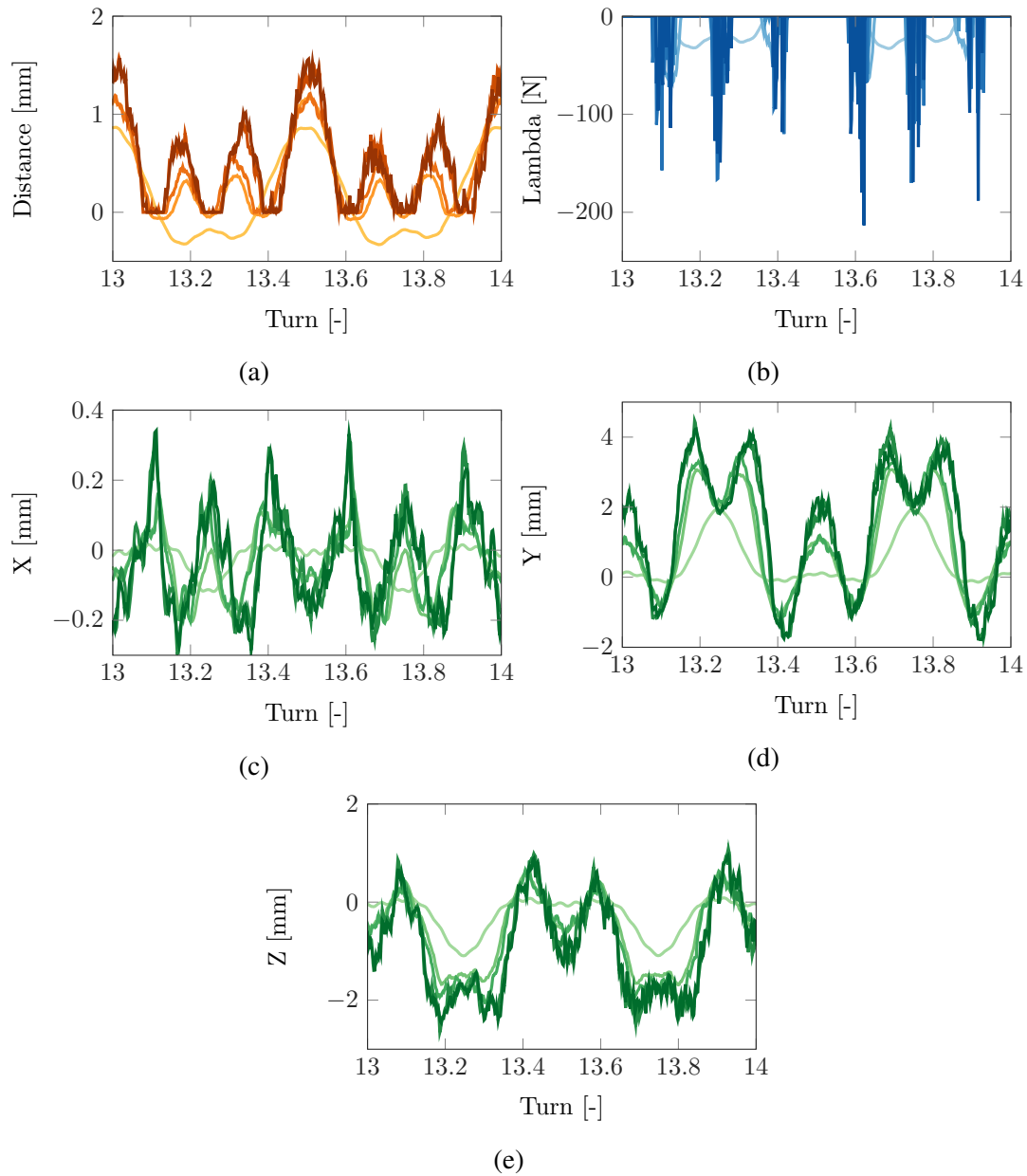


Fig. 5.9 Zoom on the distance between the trailing edge of the compressor blade and the rigid casing for different contact algorithms (a). Zoom on the contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^5$, (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

Figure 5.9 corresponds to a zoom on Figure 5.8 on one complete rotation. The six contact phases are well apparent as well as the different curves. Beside the amount of contacts between the trailing edge and the casing, the same conclusions that the ones obtained for the leading edge can be drawn regarding the penalty coefficient. In the future comparison of the two methods, $k_p = 10^8$ is the value considered.

Concerning the differences between the response of the compressor blade at the leading edge and the trailing edge, there are multiple results that can be discussed.

First concerning the positions and the duration of those contact phases at the two nodes. At the leading edge, one contact phase last in average 8.1% of the time that the blade takes to complete on rotation since there are 2 contacts per turn it is 16.2% of the time in total per rotation. The contact occurs around the two positions of minimum gap, previously represented by the red dashed line in Figure 5.5. Each contact between the trailing edge and the casing lasts in average 4.7% of the time of a rotation (28.6 % for a whole rotation). Their position have been represented in Figure 5.10.

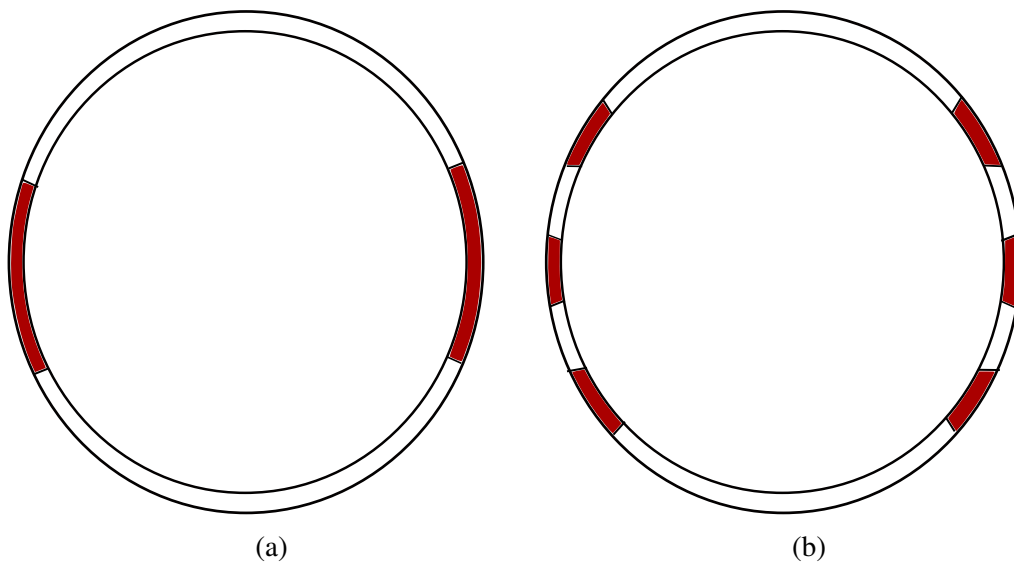


Fig. 5.10 Representation of the impacts position on the casing for the leading edge (a) and the trailing edge (b).

Second, the main source of that differences comes from the casing itself. Indeed, the differences between the gap at the trailing and leading edges, the smaller gap between the trailing edge and the casing limits the vibrations of the blade at this node. As Figures 5.6 and 5.8 are showing it, the amplitude of the displacement of the leading edge is greater than the

amplitude at the trailing edge, due to that casing limiting the free motion of the node.

Another way of looking at the result of the vibration of the compressor blade is to switch from the time domain to the frequency domain. This transformation is achieved thanks to a Fourier transform of the displacement along X . The results obtained at the leading edge and the trailing edge for the Lagrange Multipliers method and the Penalty method with $k_p = 10^8$ are given in Figure 5.11.

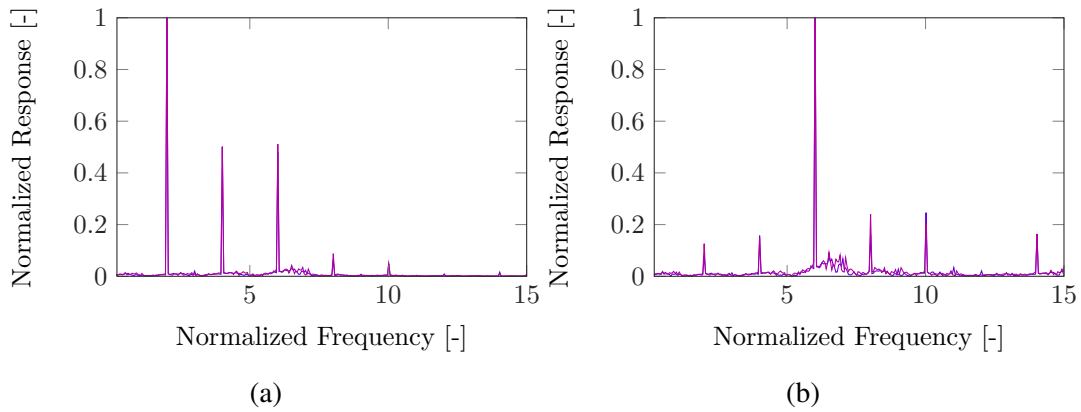


Fig. 5.11 Fast Fourier Transform of the displacement along X of the leading edge (a) and trailing edge (b) for a rigid casing. (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.

The second engine regime ($f = k\Omega$ with $k = 2$) is the one that responds more at the leading edge. On the trailing edge it is the sixth one, $k = 6$, that responds more. The harmonics of that engine regime are also responding to the excitation. This result is directly related to the amount of contact per turn. Extending the analysis of the response at the trailing edge to a larger range of rotation speed one can draw Figures 5.12 using the Lagrange Multipliers method and 5.14 using the Penalty method with, $k_p = 10^8$.

The first result to mention concerns the similitude of the results obtained with the two methods despite having significant differences in the computation process of the contact forces. This proves that both methods are viable and as long as the penalty coefficient is properly determined, they can provide identical results.

On another results concerns the increase of the value of the frequency response function (FFT) with the rotation speed. At some point it reaches a maximum value and then suddenly drops, this phenomenon has been delimited by the red dashed line in Figures 5.13 and 5.15.

This sudden increase is due to the interaction between the sixth engine regime and the associated harmonics. They are corresponding to the small peaks that can be seen between

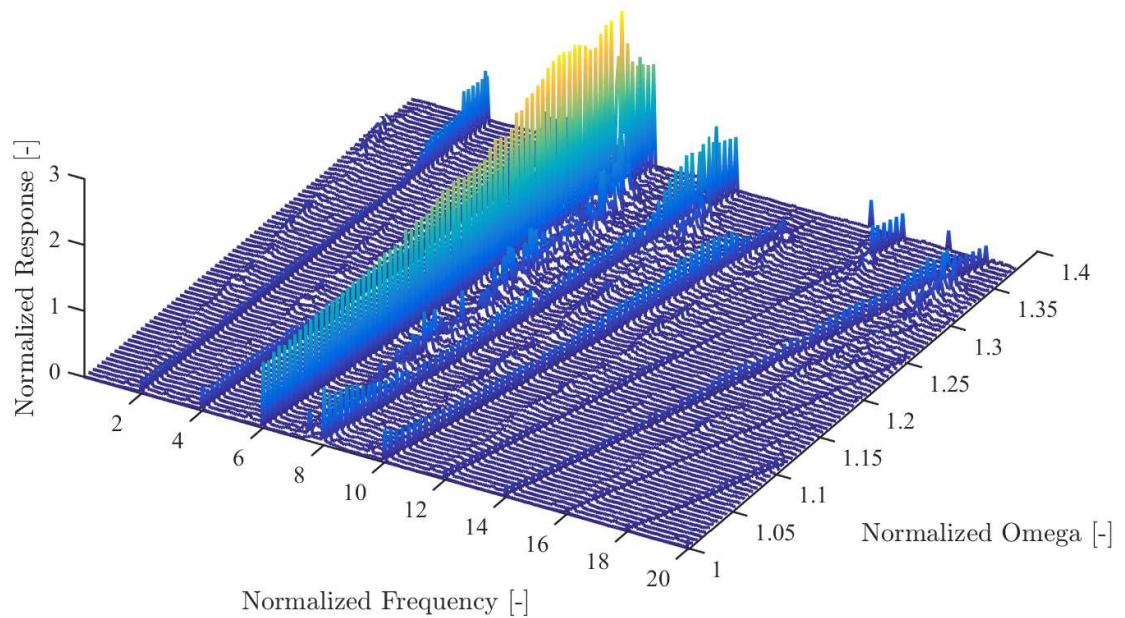


Fig. 5.12 Comparison of the FFT of the displacement along X of the trailing edge for different rotation speeds using the Lagrange Multipliers method.

the sixth and eighth engine regime in Figure 5.12 and 5.14. They can also be seen in Figure 5.13 and 5.15 as the thick yellow line near the sixth engine regime. These two figures make the interaction and the sudden drop clearly appear. This is due to an increase of the stiffness after the contact, which is translated with a shift of the response toward higher frequencies followed by a sudden jump in amplitude. This phenomenon can not be observed by looking at the leading-edge. That is why only the trailing-edge results have been presented in the end of this section.

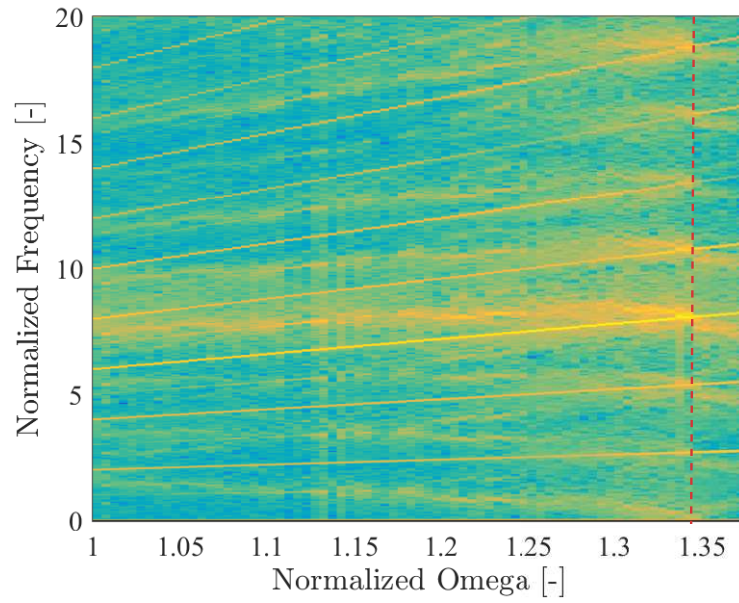


Fig. 5.13 View from the top of Figure 5.12.

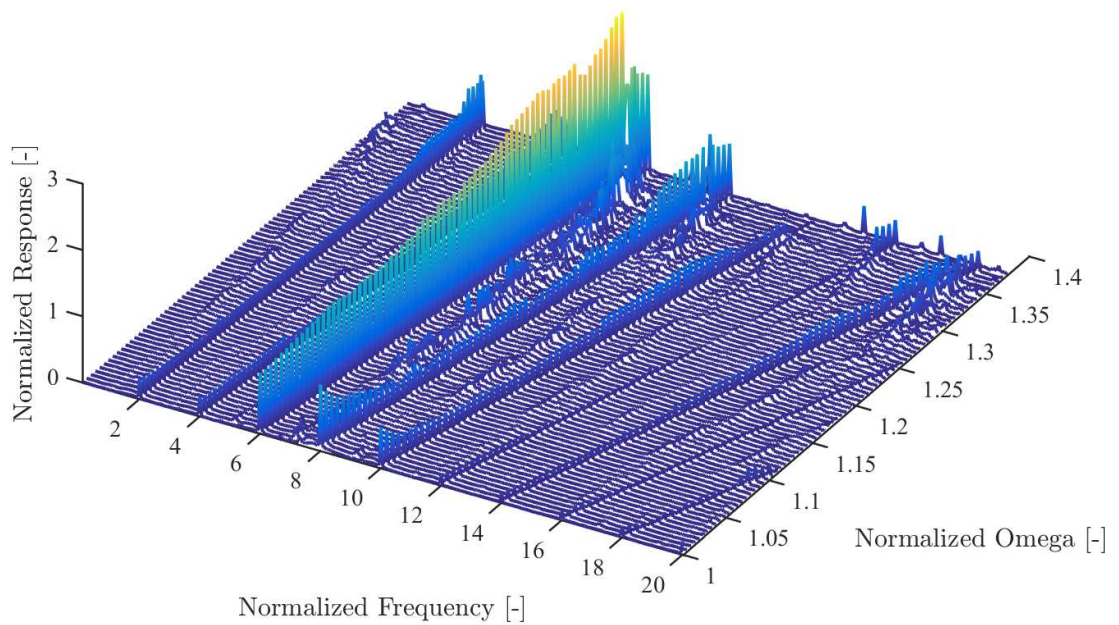


Fig. 5.14 Comparison of the FFT of the displacement along X of the trailing edge for different rotation speeds using the Penalty method with $k_p = 10^8$.

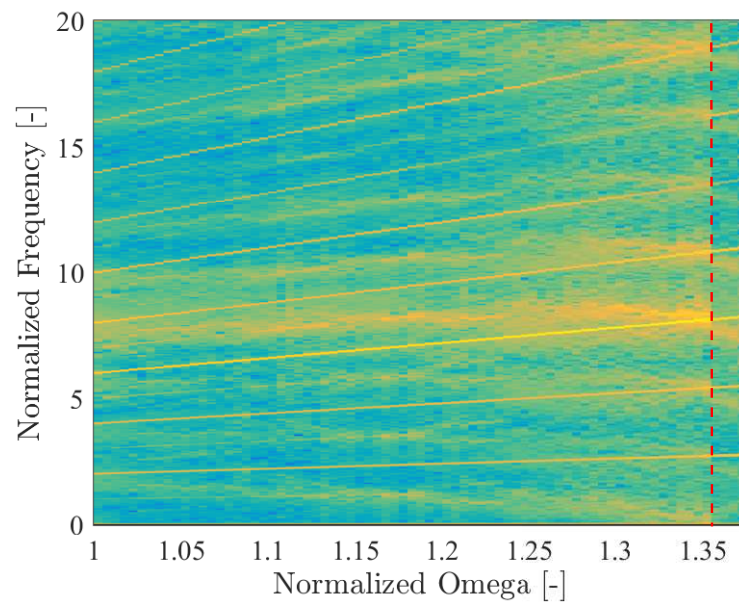


Fig. 5.15 View from the top of Figure 5.14.

5.2.2 Flexible casing

Now considering the situation where the casing is flexible. Let's mention that the casing has the same mechanical properties than the compressor blade. The results at the leading edge are corresponding to Figures 5.16, 5.17 and at the trailing edge, Figures 5.18 and 5.19.

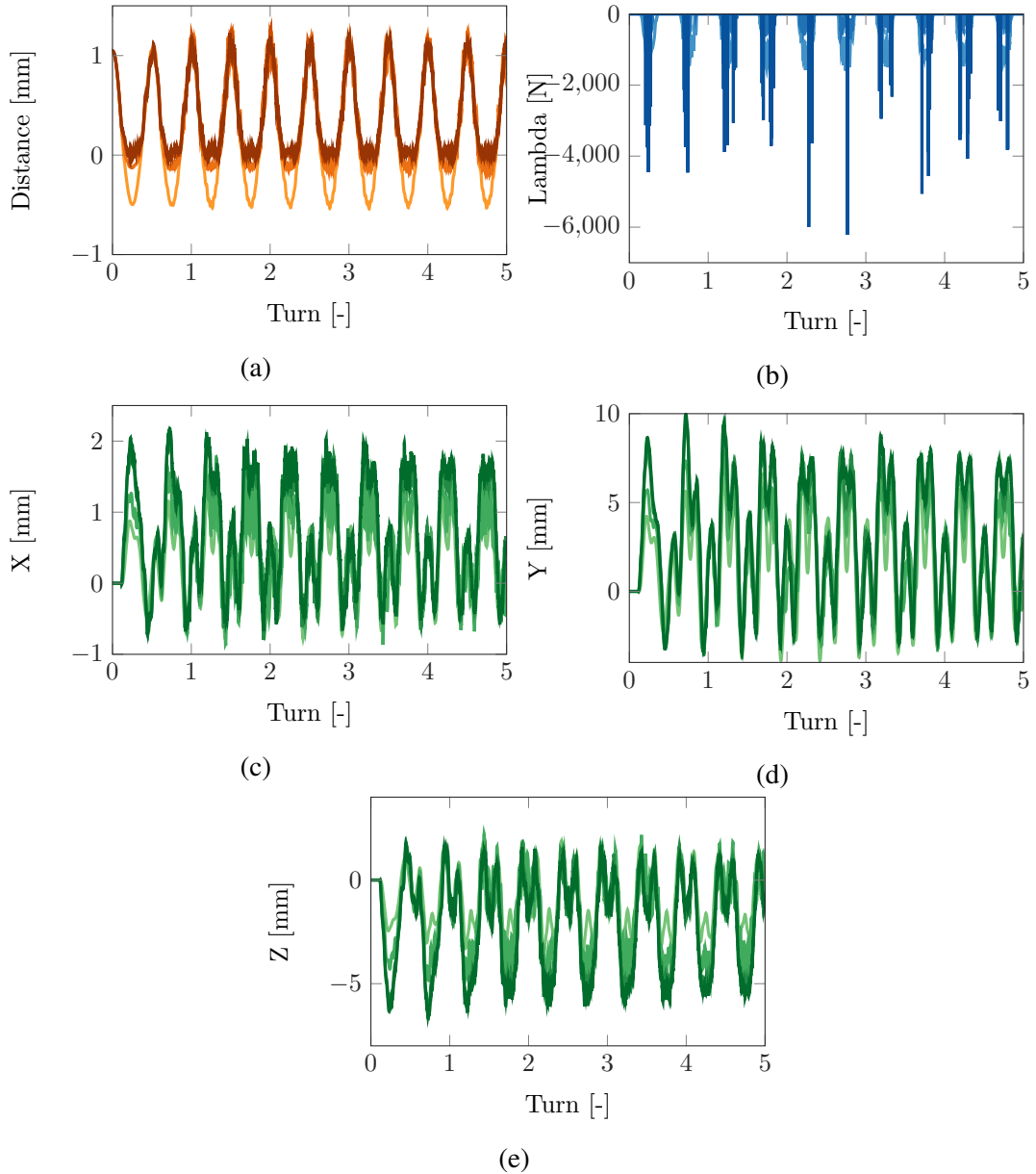


Fig. 5.16 Distance between the leading edge of the compressor blade and the flexible casing for different contact algorithms (a). Contact forces at the leading edge of the compressor blade for different contact algorithms (b). Displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

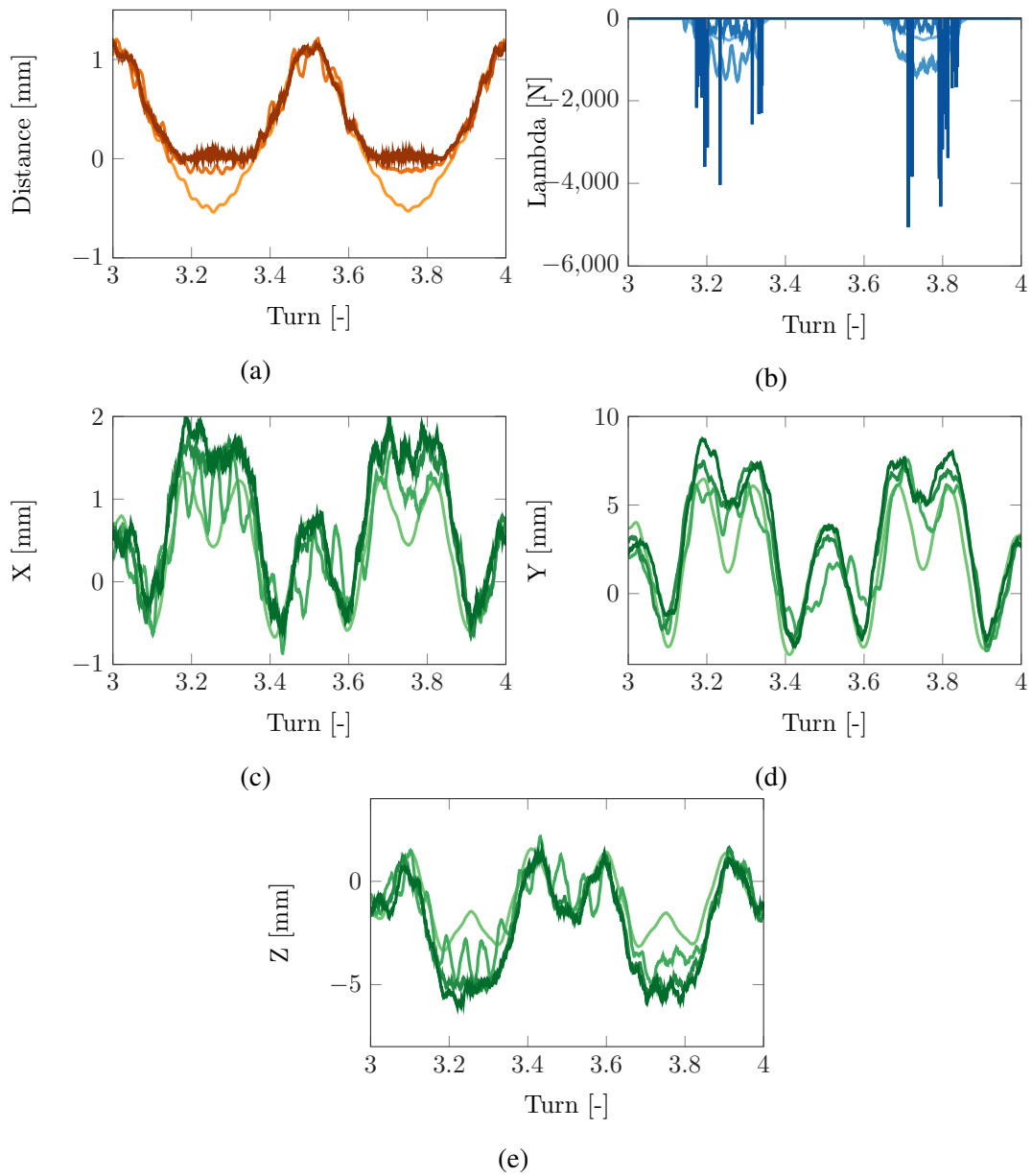


Fig. 5.17 Zoom on the distance between the leading edge of the compressor blade and the flexible casing for different contact algorithms (a). Zoom on the contact forces at the leading edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the leading edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

Flexible analysis are taking a lot more computation process than for the rigid case, the analysis is conducted on 5 turns only. Apart from that, the same parameters have been used ($h = 10^{-7}$ s, $N - R = 10^{-6}$, $C - B = 10$ for both the blade and the casing, no friction, and

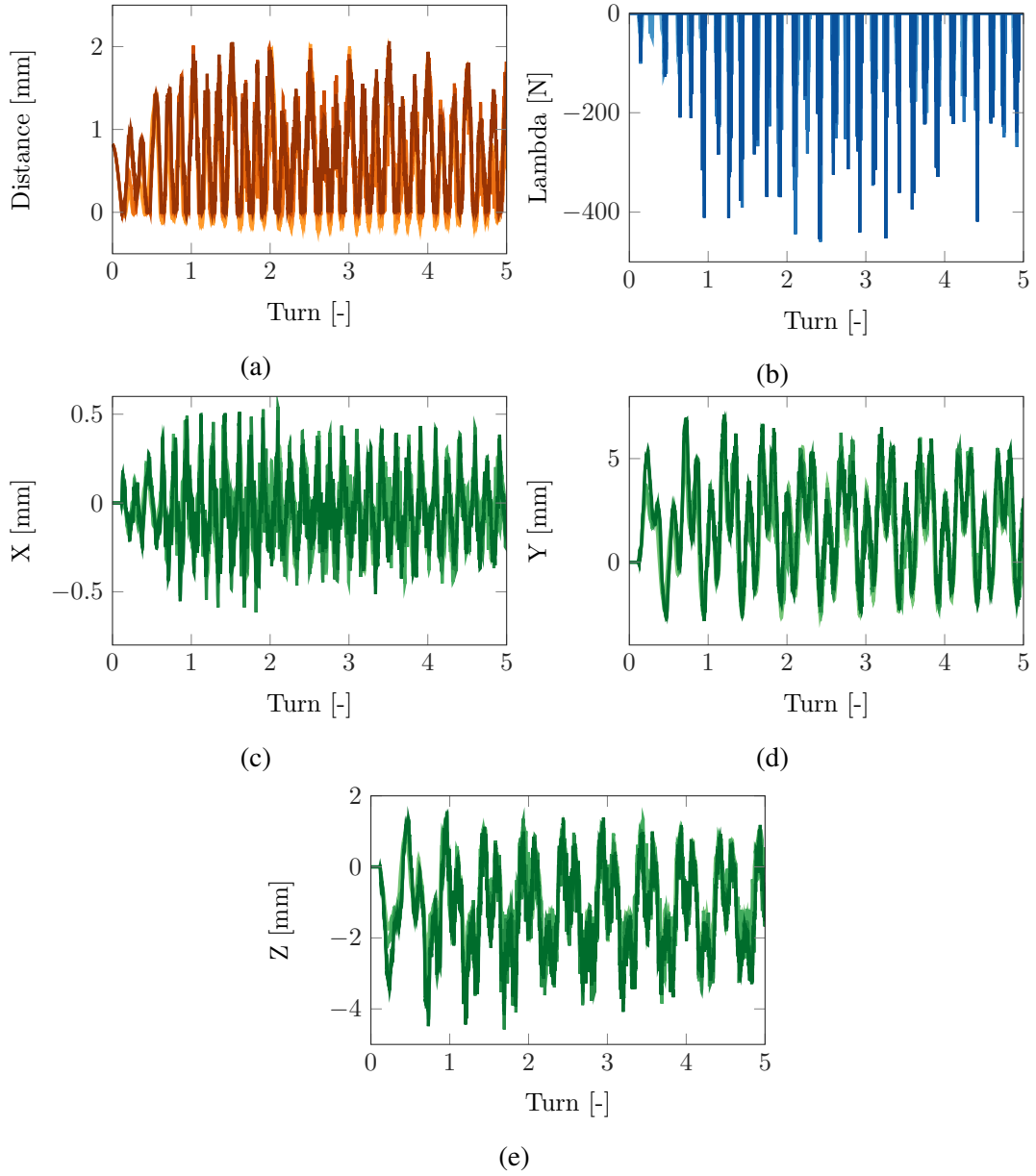


Fig. 5.18 Distance between the trailing edge of the compressor blade and the flexible casing for different contact algorithms (a). Contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (—, —, —) Penalty $k_p = 10^7$, (—, —, —) Penalty $k_p = 10^8$, (—, —, —) Lagrange Multipliers.

the same modal damping value than for the rigid casing.

The same type of results that were obtained with the rigid casing can be retrieved despite its flexibility. Two contacts per turn for the leading edge are detected while six are present at

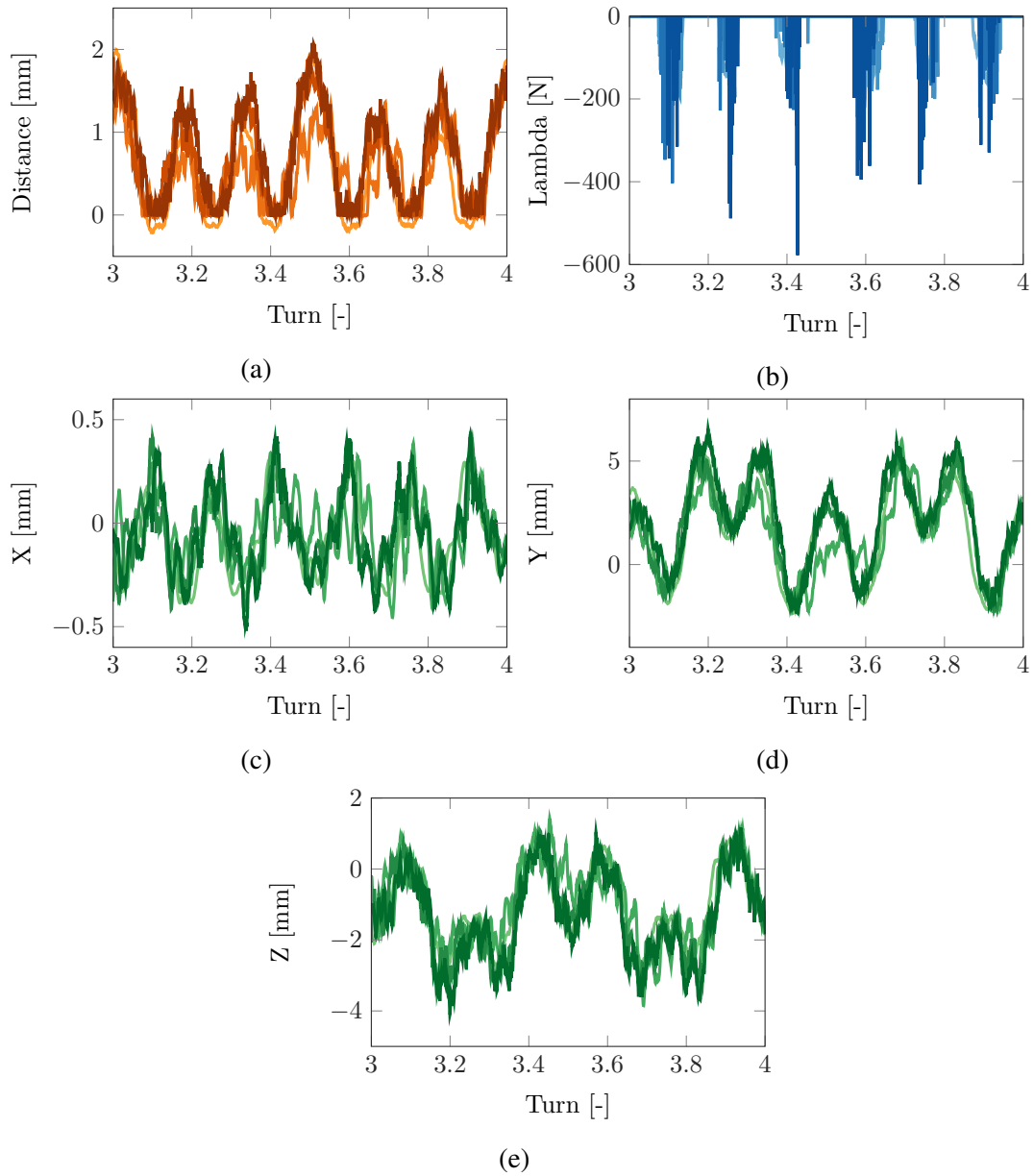


Fig. 5.19 Zoom on the distance between the trailing edge of the compressor blade and the flexible casing for different contact algorithms (a). Zoom on the contact forces at the trailing edge of the compressor blade for different contact algorithms (b). Zoom on the displacement of the trailing edge along X (c), Y (d) and Z (e) for different contact algorithms. (—, —, —) Penalty $k_p = 10^6$, (---, ---, ---) Penalty $k_p = 10^7$, (·····, ·····, ·····) Penalty $k_p = 10^8$, (---, ---, ·····) Lagrange Multipliers.

the trailing edge. Let's mention that the penalty coefficient $k_p = 10^5$ led to instabilities in the case of the flexible analysis. Globally, when looking at the zooms on one complete rotation (Figures 5.7, 5.9, 5.17 and 5.19) the responses have the same shapes between the rigid and

flexible cases. The amplitude of the vibration at the leading edge and at the trailing edge are slightly more important with the flexible casing than with the rigid one, as can be seen in the figures associated to the displacement along X , Y and Z respectively. This is due to the deformation of the casing that does not abruptly stop the motion of the blade but moves with it instead. The same conclusion regarding the penalty coefficient, k_p can be made, 10^8 is well following closely the Lagrange Multipliers results even in the flexible case.

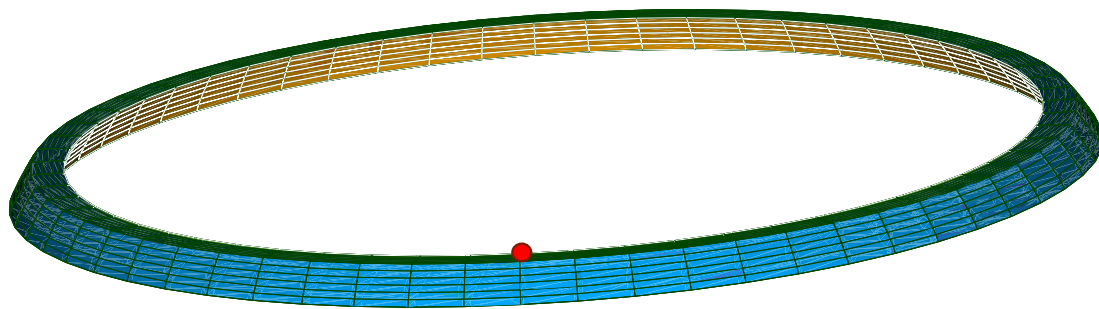


Fig. 5.20 Finite element model of the compressor blade casing with the studied nodes represented by red dots.

The displacement of the node, corresponding to the red dot in Figure 5.20, along X , Y and Z is represented in Figure 5.21 and the associated zoom on one rotation of the blade is represented in Figure 5.22.

From Figure 5.21, the transition of the displacement of the casing toward a steady state of vibration clearly appears. The frequency of the vibration of the casing is at least two times superior to the frequency of the vibration at the trailing-edge of the blade. Moreover, the amplitude is about a 100 times lower.

Lastly, one can look at the FFT at the leading and trailing edges when the casing is flexible, this corresponds to Figure 5.23. Like it was the case for the rigid casing what could have been observed in Figure 5.11, it is the same engine regime that are responding at the leading and trailing edges. On the leading edge, the highest peak is situated around the same position as the one obtained with the rigid casing, corresponding to a value of k equal to 2. At the trailing edge it is the sixth regime engine that the FFT is at its maximum.

The values of the frequencies are not exactly identical between the rigid and flexible casing. There is a shift between the responses obtained with one type of casing or the other. This is due to the interaction with the casing that modify the response frequency of the blade. In addition, the harmonics of the sixth engine regime are closer to the peak with the flexible

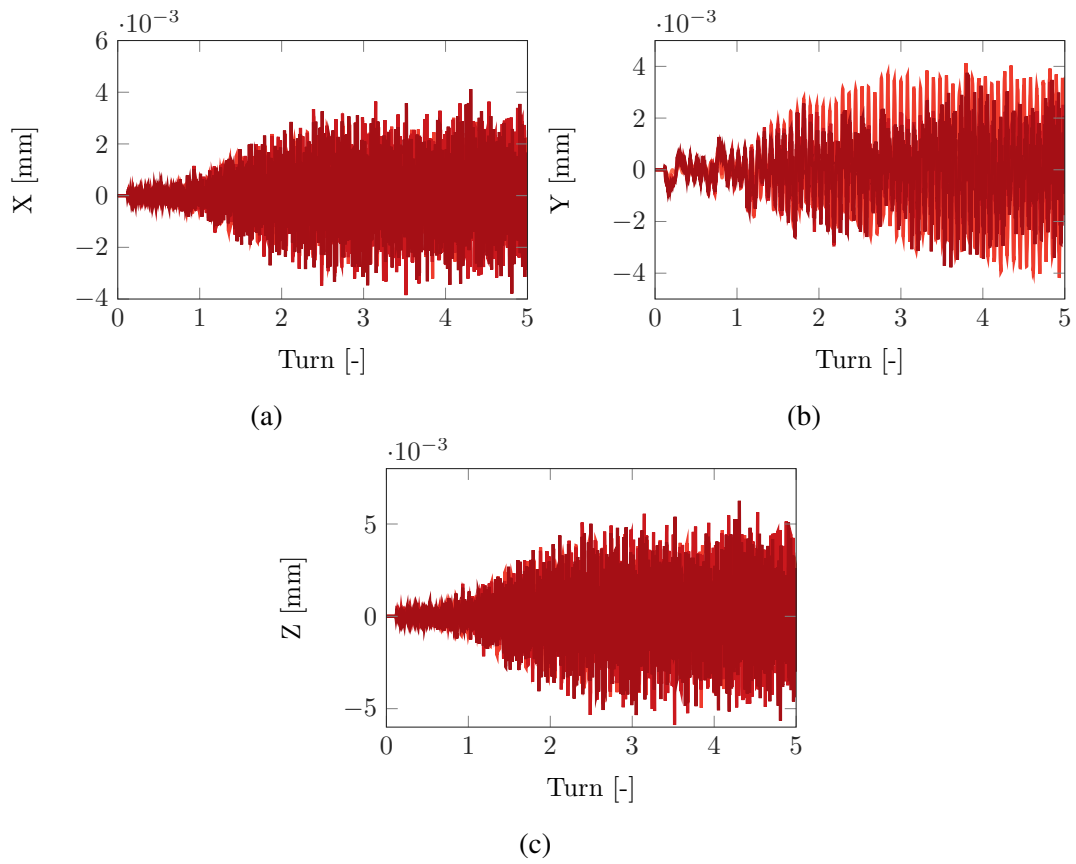


Fig. 5.21 Displacement of the node of the flexible casing along X (a), Y (b) and Z (c) for different contact algorithms. (—) Penalty $k_p = 10^6$, (—) Penalty $k_p = 10^7$, (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.

casing that they were with the rigid one. This means that the jump will most likely happen before what have been found for the rigid casing.

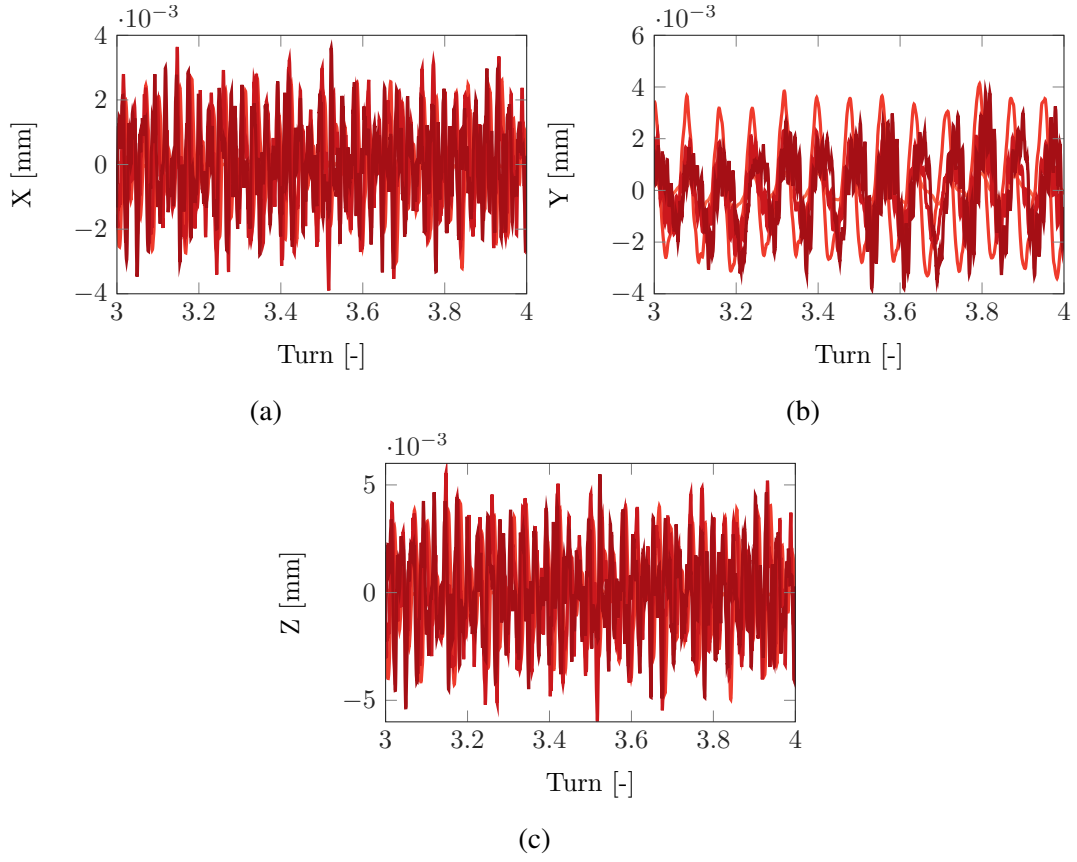


Fig. 5.22 Zoom on the displacement of the node of the flexible casing along X (a), Y (b) and Z (c) for different contact algorithms. (—) Penalty $k_p = 10^6$, (—) Penalty $k_p = 10^7$, (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.

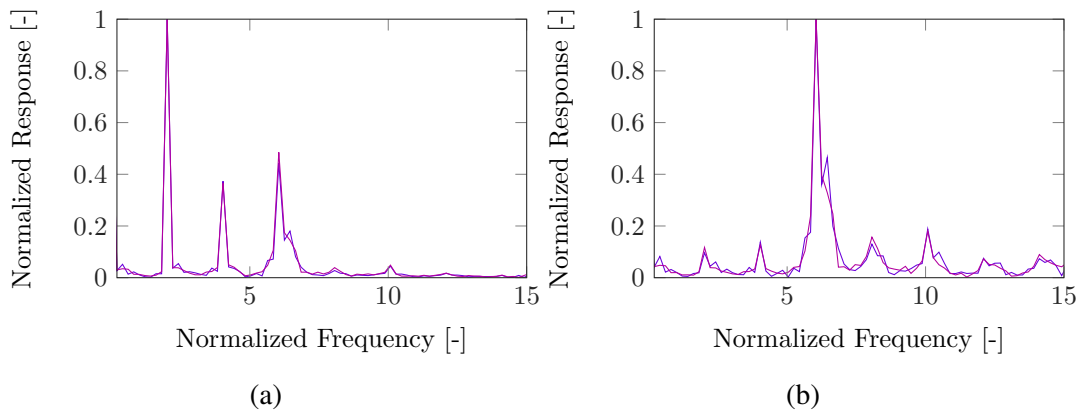


Fig. 5.23 Fast Fourier's Transform of the displacement along X of the leading-edge (a) and trailing-edge (b) for a flexible casing. (—) Penalty $k_p = 10^8$, (—) Lagrange Multipliers.

Conclusion

Throughout this chapter, a complete and detailed analysis of the interaction of a turbine blade has been carried out.

In the first section, a general presentation of the test case is given. In addition to that the effectiveness of reduction method thanks to the Craig-Bampton method have been demonstrated on the case of the compressor blade.

In the second section, the analysis of 20 full rotations of the blade has been performed. Interesting results such as the fact that different engine regimes were responding depending on what node of the blade were observed. Furthermore, the Penalty method and the Lagrange Multipliers method have first been compared at the same rotation speed. Despite very different contact forces the responses were almost similar for the two methods, it proved that they are both valid methods. The analysis has then been carried in the frequency domain with the study of the FFT for different rotation velocities. A non linear phenomenon known as a jump of the FFT has been highlighted. Due to the contact there is a stiffening of the blade that induces that non linearity in the response.

The last section dealt with the analysis of the blade inside a flexible casing. The analysis of the FFT, at the same rotation speed that the one considered in the case with the rigid casing, allowed to highlight the possible interaction between the response of the blade and of the casing.

In the end, this chapter validated the efficiency of the numerical tool regarding the simulations of complex structures such as the turbine blade. It allows to retrieve interesting results such as the non linear jump of the FFT and the difference of vibrations between the trailing and leading edges.

Chapter 6

Conclusions

6.1 General Conclusions

This master thesis offers the presentation of a laboratory numerical tool to model and simulate contact interactions between multiple structures. As mentioned in the introduction, the interest for non-linear interactions keeps on growing. The contact interactions can be critical in a lot of applications. It is to respond at that interest that this software have been developed. It have been coded in the main objective of bringing an accessible and yet efficient software that can simulate the contact of any object without any restriction on the geometry. A tool that can be freely used without any commercial constraint, providing multiple options for the resolution of the contact. A tool that can easily be upgraded with additional functions providing new approaches and extended analysis.

In the first chapter, a detailed presentation of the theoretical background have been given. It starts with an overall introduction to the time integration method and the Newmark schemes. Among all the schemes presented the central difference suits the best the contact analysis. The reduction methods have then been introduced. The theory behind the construction a b-spline surface have been presented. This allowed the construction of continuous surfaces despite the discrete set nodes provided by a mesh. Finally, the algorithm to determine the closest point on this surface was presented and the two algorithms, the Lagrange Multipliers and Penalty method were described. This purely theoretical chapter, despite not containing any results, provides a complete introduction to the concepts that lead to a full description of a contact simulation.

The second chapter was dedicated to the presentation of the developed tool on a practical point of view. Due to its fast computation processing, the core of the code have been

programmed in C++. On the contrary, the initialization of the tool was made via Python, an user friendly and accessible language that has the advantage of share its structure with the open-source finite element software used for construction of the structures model. This common ground made the extraction of the useful data such as the mass and stiffness matrices easier.

The third chapter concerns the validation of the different aspects, numerical and physical, of the numerical tool. A cube falling between the two rings case helped to highlight a lot of interesting aspect of the tool. The symmetry of the case was retrieved in the results. The dependencies of the response with respect to the mesh have been presented. The coarser the mesh the less accurate the results. This is the conclusion that have been drawn from the obtained results. However, they are converging toward the same results the more points are added to the mesh. The convergence of the results have also been demonstrated with respect to the reduction parameter, the Newton-Raphson parameter and the time-step. These studies allowed to determine the right set of parameters for the physical studies. The physical parameters are variables like the consideration of the flexibility of the rings allowing deformations, the introduction of damping in the system and consider the effects of the friction. Interesting results were obtained, for instance, the deformation of the rings allowing the cube to go through them, or the friction that blocks the cube between them. Many different behaviours proving the viability in the representation of physical behaviours of the developed tool. The final part of this chapter concerned the comparison of the Lagrange Multipliers and the Penalty method. The impact of a poor decision on the penalty coefficient have been highlighted.

Finally, in the fourth and last chapter, the analysis of a compressor blade in rotation entering in contact with its casing has been addressed. The impact of the reduction method has first been presented, and the results proved the advantages of it on large structure and complex finite element model. A first analysis allowed to determine the right value of the penalty coefficient in order to perform a detailed analysis. Interesting results have been observed when the response at the leading edge and the response at the trailing-edge were compared. Two contacts were detected at the leading edge while six were observed at the trailing edge. Once the analysis is transposed to the frequency domain, the associated response respectively appeared on the second and sixth engine regime. In addition to that, the harmonics of the sixth engine regime clearly appeared on the frequency analysis of the trailing edge. Increasing the rotation velocity, the interaction between the response and the harmonics amplified the frequency response up to a certain point where the harmonics and

the response are fused together. After what the response suddenly dropped. This nonlinear behaviour retrieved thanks to the developed tool can be considered as a last validation that proves its robustness. Lastly, the analysis have been conducted with a flexible casing where the interactions between the casing and the blade have been noticed in the frequency domain.

6.2 Perspectives

The development of this numerical tool started from nothing, it is obvious that such a complex topic as the contact interactions can not entirely be modelled within the scope of an internship. But this tool has been designed to grow and to converge toward a complete software providing a great range of options to model with fidelity the reality of contact behaviours.

Two types of improvements can be distinguished. The first category lies in the programming point of view. In order to improve the performances of the already existing code some upgrades can be made. First, regarding the computation including matrices. The matrices have been created as an object in the developed tool, but another approach consists in the usage of libraries of linear algebra such as BLAS (Basic Linear Algebra Subprograms) [23]. The BLAS library offers fast and optimized computation for matrices.

Another improvement that can be include in the already existing code concerns the parallelization of the calculations. The parallel computing is a type of computation that allows to run simultaneously on different core of a same computer multiple tasks. This is possible using libraries such as OpenMP [8] or MPI [20]. This method divides the computation time per amount of core associated to the simulation. However, only some parts of the code can be put in parallel. For instance, the evaluation of the distance between the nodes and the surface is computed for each nodes independently. Assuming that there are as many cores as there are of nodes, it would have been possible to assign at each core the task of computing the distance. Since this operation is repeated at every time-steps, the improvement can be significant.

Regarding the growth of the developed tool on a physical representation side, some future developments can be briefly presented as well. During the contact it is possible that a one of two structure wore. This aspect of wear represents the natural progression of the developed tool. Also, contact events may lead to high temperature increases, that cannot be neglected for some applications. Therefore, taking into account thermal effects in the model could also

be an interesting future development for the code.

References

- [1] Gnuplot. <http://www.gnuplot.info/>.
- [2] Gnuplot-iostream. <http://stahlke.org/dan/gnuplot-iostream/>.
- [3] A. Valdescault, A. Batailly, S. J. (2012). Interpolation et approximation de données à l'aide de courbes et surfaces paramétriques de type b-splines.
- [4] A. Batailly and Legrand, M. (2015). Unilateral contact induced blade/casings vibratory interactions in impellers: Analysis for flexible casings with friction and abradable coating. *Journal of Sound and Vibration*, 348:344–364.
- [5] Booch, G. (1986). Object-oriented development. *IEEE transactions on Software Engineering*, (2):211–221.
- [6] Boor, C. D. (1972). On calculating with b-splines. *Journal of Approximation Theory*, 6:50–62.
- [7] Cerfontaine, B., Dieudonné, A.-C., J.P. R., Collin, F., and Charlier, R. (2015). 3d zero-thickness coupled interface finite element: Formulation and application. 69:124–140.
- [8] Chandra, R., Dagum, L., Kohr, D., Maydan, D., Menon, R., and McDonald, J. (2001). *Parallel programming in OpenMP*. Morgan kaufmann.
- [9] Craig, R. R. and Bampton, M. C. C. (1968). Coupling of Substructures for Dynamic Analyses. *AIAA Journal*, 6(7):1313–1319.
- [10] de France, E. (1989–2018). Finite element *code_aster*, analysis of structures and thermomechanics for studies and research. Open source on www.code-aster.org. Accessed : 16-08-2016.
- [11] Gavin, H. P. (2016). Numerical integration in structural dynamics. Departement of Civil and Environmental Engineering Duke University.
- [12] Golinval, J. (2018). Theory of vibration. Chap 07 Slide 24.
- [13] G rardin, M. and Rixen, D. (2015). *Mechanical Vibrations, Theory and Application to structural dynamics*. Wiley.
- [14] GUYAN., R. J. (1965). Reduction of stiffness and mass matrices. *AIAA Journal*, Vol. 3:380–380.

- [15] Newmark, N. (1959). A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*.
- [16] N.J. Carpenter, R. T. and Katona, M. (1991). Lagrange constraints for transient finite element surface contact. *International Journal for Numerical Methods in Engineering*, 32:103–128.
- [17] Peter Deuffhard, Rolf Krause, S. E. (2006). A contact-stabilized newmark method for dynamical contact problems. 6.
- [18] Schoenberg, I. J. (1973). *Cardinal spline interpolation*, volume 12. Siam.
- [19] Shimizu, T. and Sano, T. (1995). An application of a penalty method contact and friction algorithm to a 3-dimensional tool surface expressed by a b-spline patch. *Journal of Materials Processing Technology*, 48:207–213.
- [20] Snir, M., Otto, S., Huss-Lederman, S., Dongarra, J., and Walker, D. (1998). *MPI—the Complete Reference: The MPI core*, volume 1. MIT press.
- [21] Unknown. Basic components of a gas turbine engine. https://en.wikipedia.org/wiki/File:FAA-8083-3A_Fig_14-1.PNG. Accessed : 16-08-2016.
- [22] Unknown. Benchmark c++ vs python. <https://benchmarksgame-team.pages.debian.net/benchmarksgame/faster/gpp-python3.html>. Accessed : 16-08-2016.
- [23] Unknown. Blas (basic linear algebra subprograms). <http://www.netlib.org/blas/>. Accessed : 16-08-2016.
- [24] Unknown. Cad of an aircraft engine. <https://grabcad.com/library/jet-engine-13>. Accessed : 16-08-2016.
- [25] Unknown. Compiled language vs interpreted language. https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zappldev/zappldev_85.htm. Accessed : 16-08-2016.
- [26] Y.T.Feng (2005). On the central difference algorithm in discrete element modelling of impact. *International journal for numerical methods in engineering*.